International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Special Issue 2, April 2014)
National Conference on Computing and Communication-2014 (NCCC'14)

Maximizing Coverage Problem Using Genetic Algorithm in Wireless Sensor Network

M. Mohamed Aadhil[1], J. Roselin, M.E.[2]
[1]PG Scholar, [2]Assistant Professor, Department of Computer Science and Engineering,Regional Centre of Anna University, Tirunelveli
aadhilforu@gmail.com1, roselin_js@rediffmail.com2

***Abstract*- Sensor networks, which consist of sensor nodes each capable of sensing environment and transmitting data, have lots of applications in battlefield surveillance, environmental monitoring, industrial diagnostics, etc. Coverage which is one of the most important performance metrics for sensor networks reflects how well a sensor field is monitored. In this paper, we introduce the maximum coverage deployment problem in wireless sensor networks. Random deployment is the simplest way to deploy sensor nodes but may cause unbalanced deployment and therefore, we need a more intelligent way for sensor deployment. Normalization plays a role of reducing the search space to another one of less size. Based on this property, we propose an efficient genetic algorithm using a novel normalization method. The proposed genetic algorithms could be further improved by combining with a well-designed local search. The performance of the proposed genetic algorithm is shown by a comparative experimental study. When compared with random deployment and existing methods, our genetic algorithm was not only about twice faster, but also showed significant performance improvement in quality.**

***Keywords*-- Coverage, Maximum Coverage, Genetic Algorithm, Normalization.**

I. INTRODUCTION

A Wireless Sensor Network (WSN) can be composed of homogeneous or heterogeneous sensors [1], which possess the same or different communication and computation capabilities, respectively. Although some works consider heterogeneous sensors, many existing works investigate node placement in the context of homogeneous WSNs. Less complexity and a better manageability are the most beneficial effects of homogeneity. Therefore, we consider homogeneous nodes in WSNs. These nodes can be deployed over a network in random or deterministic fashion.

While the random node deployment is preferable in many applications, if possible, other deployments should be investigated since an inappropriate node deployment can increase the complexity of other problems in WSNs. Research in the field of mobile wireless sensor networks is motivated by the need to monitor hostile environments such as wild fires, disaster areas, toxic regions or battlefields, where static sensor deployment cannot be performed manually. In these working settings, sensors may be dropped from an aircraft or sent from a safe location. Mobile sensors can dynamically adjust their position to improve the coverage [12] with respect to their initial deployment.

Although the price of sensors gradually decreases as the sensor technology evolves, cost considerations still make it necessary to cover a wide range of a target area with a minimum number of sensors, which can be accomplished by efficient deployment of the sensors. For a small number of sensors, manually placing them after considering their sensing range can be effective. But as most sensor network applications deal with a large number of sensors.

Random deployment is the simplest way to deploy sensor nodes but may cause unbalanced deployment and therefore, we need a more intelligent way for sensor deployment**.** Covering as broad an area as possible is still necessary and achievable using sensor networks. In real world, it frequently appears that the sensor field is too large to fully cover with a limited number of sensors. This problem cannot be solved by deterministic methods such as the circle packing algorithm because the sensing range may not be equal. We define the problem formally and propose an efficient genetic algorithm for solving the problem.

A Monte Carlo method is adopted to design an efficient evaluation function, and its computation time is decreased without loss of solution quality using a method that starts from a small number of random samples and gradually increases the number for subsequent generations. A well designed Local search helps to get the optimized result.

In this paper, we formally defined the maximum coverage sensor deployment problem (MCSDP), which frequently aroused in real-world applications. We analyzed the properties of the problem space and tried to find good sensor deployments using Monte Carlo and novel genetic algorithms.

II. RELATED WORK

A. Assumptions

Sensors have size, weight, and cost restrictions, which impact resource availability. They have limited battery resources and limited processing and communication capabilities. As replacing the battery is not feasible in many applications, low power consumption is a critical factor to be considered, not only in the hardware and architectural design, but also in the design of algorithms and network protocols at all layers of the network architecture. Therefore maximizing the network lifetime is an important network design objective. Using a minimum number of sensors is another clear objective, especially in a deterministic node deployment approach.

Sensor coverage models measure the sensing capability and quality by capturing the geometric relation between a point and sensors. Among them, the Boolean disk coverage model might be the most widely used sensor coverage model in the literature. The coverage function [13] of the model is given by

$$f\big(d(s,x)\big) = \int \begin{matrix} 1\ if\ d(s,x) \le rs \\ 0 \quad\quad otherwise \end{matrix}$$

Where d(s, x) is the Euclidean distance between a sensor s and a point x, and the constant rs is called sensing range. Indeed this function defines a disk centered at the sensor with the radius of the sensing range. This model is an omnidirectional coverage model. All points within such a disk have a coverage measure of 1, and are said to be covered by this sensor. All points outside such a disk have a coverage measure of 0, and are said not covered by this sensor.

The sensing range rs is used to characterize the sensing capability of a sensor. In general, different sensor types are assumed to have different sensing ranges.

According to the subject to be covered, coverage in sensor networks can be classified into three types: area coverage, barrier coverage, and target coverage. In this paper, we adopt the Boolean disk coverage model and deal with area coverage problem, which equally treats every point in the sensor field. In this paper, as a new trial, we focus on the problem of maximizing the covered area of the sensor field with a given number of sensors. This problem can not only be directly used to solve previous area coverage problem that minimizes the number of sensors to cover the whole sensor field by adjusting the number of sensors in a binary search manner, but can also be transformed into other types of coverage problems, e.g., barrier coverage problem in, which constructs a barrier that maximizes the coverage of vehicle paths across the sensor field with a given number of sensors. Although we assume a specific coverage problem, the proposed methodology is easily applicable to other types of coverage problems by just changing the evaluation function.

The coverage algorithms proposed are either centralized or distributed. In distributed algorithms, the decision process is decentralized. By distributed and localized algorithms, we refer to a distributed decision process at each node that makes use of only neighborhood information, within a constant number of hops. Because the WASN has a dynamic topology and needs to accommodate a large number of sensors, the algorithms and protocols designed should be distributed and localized, in order to better accommodate a scalable architecture.

B. Problem Formulation

We define the maximum coverage sensor deployment problem (MCSDP) in the following way. There are k types of static sensors, and each type of sensors can cover a given area with an arbitrary fixed radius r1, r2. . . rk. The total number of sensors is n , and assuming that there is at least one sensor for each type of sensors, sensors for type i are numbered from ni to ni+1 − 1 with n1 = 1 and nk+1 = n + 1. That is, for k + 1 numbers n1(= 1) < n2 < · · · < nk+1(= n + 1), there are ni+1 − ni sensors for sensor type i, where i = 1, 2, . . . , k.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Special Issue 2, April 2014)
National Conference on Computing and Communication-2014 (NCCC'14)

The objective of the problem is to find locations (x1, y1), (x2, y2), . . . , (xn, yn) for all n sensors that produce the maximum coverage for a given domain A. Consider a circle Cr(x, y) for which the center is (x, y) and the radius is r, i.e., Cr(x, y) := {(v,w) ∈ R2 : (v − x)2 + (w − y)2 ≤ r2}, the problem can be formalized as follows:

$$\text{Maximize area } \left(\bigcup_{i=1}^{k} \bigcup_{j=n_i}^{n_{i+1}-1} C_{r_i}(x_j, y_j) \cap A\right)$$

C. Test Data Set

We assume that the domain *A* to be covered with sensors is a square region in 2-D Euclidean space, [0, 100]×[0, 100]. We used 3 types of sensors for which the detection radius for each sensor type is r_1, r_2, and r_3, respectively. The values of radii are determined by the criteria that $r_1 = 0.8 \times r_2$ and $r_2 = 0.8 \times r_3$. The number of sensors for each type is determined to match the given tightness ratio α, i.e., to make the total of each sensor's detection area be a product of α and the area of given domain *A*. In this manner, 15 test instances given in Table I were generated. The total number *n* of sensors varies from 17 of instance S1-0.7 to 130 of instance S5-0.9. Main parameters in our test data generation are modeled from literature including real-world sensor information. Lamm [28], Srour [40], and Seo *et al.* [37] dealt with 3 types of sensors named acoustic sensor, seismic sensor, and forwardlooking infrared radar (FLIR), which are used in battlefield surveillance of the U.S. Army. The detection radius of seismic sensor is about 0.8 times that of acoustic sensor. Similarly, the detection radius of FLIR is about 0.8 times that of seismic sensor. For the tightness ratio α, we referred to the work of Zou and Chakrabarty [56], [57]. They used just one type of sensors and determined the number of sensors with the tightness ratio 0.7. We extended their test data with more sensor types (3 types) and more various tightness ratios (0.8 and 0.9) including 0.7 because the price of sensors is expected to be lowered as the technology for manufacturing sensors advances. Given a budget, we will be able to use more sensors to cover a given domain. So additionally we considered larger tightness ratios (0.8 and 0.9) than 0.7 for an extended study.

III. PROPOSED SYSTEM

A. Genetic Framework

Genetic algorithm [14] is a computational model that simulates the process of genetic selection and natural elimination in biological evolution. It has been widely used to solve the combinatorial and non-linear optimization problems with complex constraints or non-differentiable objective functions. The computation of genetic algorithm is an iterative process towards achieving the global optimality. During the iterations, candidate solutions are retained and ranked according to their quality. A fitness value is used to screen out unqualified solutions. Genetic operations of crossover, mutation, translocation, inversion, addition and deletion are then performed on those qualified solutions to create new candidate solutions of the next generation. The above process is carried out repeatedly until certain stopping or convergence condition is met for simplicity, a maximum number of iterations can be chosen to be the stopping condition. The variation difference of the fitness values between two adjacent generations may also serve as a good indication for convergence. To utilize the genetic algorithm method, various parts of the SDP must be mapped to the components of the genetic algorithm as will be shown in this section.

We describe the genetic framework [13] we use for our study. Let *N* be the population size. A collection of *N/*2 pairs is randomly composed, and crossover and mutation are applied to each pair, generating *N/*2 offspring. Parents and newly generated offspring's are ranked, and the best *N* individuals among them are selected for the population in the next generation. In all our experiments, a population size of 50 was used (*N* = 50). Our GAs terminate after 1,000 generations. As crossover and mutation operators, BLX-α and Gaussian mutation (described below) are used, respectively.

• *BLX-α:* Offspring $z = (z1, z2 \ldots z2n)$ is generated from parents $x = (x1, x2, \ldots, x2n)$ and $y = (y1, y2, \ldots, y2n)$, where zi is uniformly randomly chosen from the interval $[\min\{xi, yi\} - \alpha I, \max\{xi, yi\} + \alpha I]$, where $I = |xi - yi|$. Since the value 0.5 for α is widely used, we also used the same value in our GAs.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Special Issue 2, April 2014)
National Conference on Computing and Communication-2014 (NCCC'14)

• *Gaussian mutation:* The *i*-th gene xi of an individual is mutated by $N(0, \sigma i)$ with mutation rate pm, where $N(0, \sigma i)$ is an independent random Gaussian number with the mean of zero and the standard deviation of σi. In our GAs, σi is fixed to 50 - a half of width of a given domain $A = [0, 100] \times [0, 100]$ -and pm is fixed to $0.1/n$, where n is the total number of sensors.

B. Evaluation

The evaluation function of the MCSDP corresponds to calculating the total area covered with emplaced sensors. That is, the evaluation value is the area of union of circles centered around each sensor's location, and the radius is each sensor's detection range. Thus, we practically evaluated a given deployment by a Monte Carlo simulation method. Monte Carlo methods perform repeated random sampling to compute results. Let *X be* a covered area

$$\text{Maximize area } \left(\bigcup_{i=1}^{k} \bigcup_{j=n_i}^{n_{i+1}-1} C_{r_i}(x_j, y_j) \cap A)\right)$$

$$\text{AREA (X)} = \int_A Ix(x,y)\,dxdy$$

C. Normalization

To verify that the normalization [23] method actually works on the MCSDP, we performed experiments with three rearrangement strategies. Assume that one parent x is $((p1, q1), (p2, q2)\ldots (p\ n, q\ n))$ And the other parent y is $((r1, s1), (r2, s2). . . (r\ n, s\ n))$.

• *RAND:* Without rearrangement, the original arrangement of the second parent is chosen before recombination.

• *MINDIST:* We rearrange the genes of the second pare nt to minimize the distance sum, i.e., the value of

$$\sum_{i=1}^{n} \sqrt{d((pi,qi),(ri,si))}$$

If we adopt the Euclidean distance as a distance for 2-D coordinates, the value becomes

$$\sum_{i=1}^{n} \sqrt{(pi - ri)2 + (qi - si)2}.$$

• *MAXDIST:* We rearrange the genes of the second parent to maximize the distance sum, i.e., the value of

$$\sum_{i=1}^{n} \sqrt{d((pi,qi),(ri,si))}$$

Here, MINDIST method is the closest to the meaning of normalization since both parents can be said to be similar to each other if the distance between them is small. Moreover, the minimum distance sum in the MINDIST method becomes a metric on the phenotype space $G/\sim$ by the following proposition.

Normalization Steps:

1. N solutions are randomly generated. (N is100.)

2. $N/2$ pairs of solutions are produced.

3. Rearrangement is applied to the second parents by the three aforementioned methods.

4. Crossover operator (BLX-0.5) is applied to each pair the first parent and the rearranged second parent.

5. The average and the standard deviation of the coverage of $N/2$ offspring are calculated.

MINDIST method is superior to the others as we expected, and we can expect good performance of GAs when using the MINDIST method as normalization

IV. Experiment

We conducted experiments on 15 instances presented in Table I (see Section II-C) to verify the performance of the proposed GAs. Tables IV and V show the results. Thirty trials were performed for each method and the results were averaged over the trials. "Time" is the average CPU seconds that were taken to perform 30 trials on an Intel Xeon CPU 2.4 GHz. In the table, PGA means a pure genetic algorithm without the proposed normalization and speed-up techniques. Monte Carlo method was used with a fixed number of random samples (100,000) to evaluate the solutions. To verify the effectiveness of the GA, we compared the results with a multistart method using randomly-deployed solutions [44]. As a single run, we repeatedly sampled a number of randomly-deployed solutions in the given domain $A = [0, 100] \times [0, 100]$ for an amount of time similar to the GA, and chose the best one. RANDOM in the table is the result. When we compare the results of RANDOM and PGA, PGA showed significantly better performance than RANDOM (see the results of one-tailed *t*-tests in PGA). It can be seen that GAs performed well on the MCSDP.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (ISSN 2250-2459, ISO 9001:2008 Certified Journal, Volume 4, Special Issue 2, April 2014)
National Conference on Computing and Communication-2014 (NCCC'14)

GA using Monte Carlo evaluation with an increasing number of random samples (called MGA) mentioned in Section III-B was also tested. In MGA, the initial number of random samples is 10,000 and increased by 10,000 per 100 generations. MGA was about two times faster than PGA, so the computational cost could be largely reduced using such a method. MGA uses the same framework as PGA except its evaluation time was reduced with the proposed speed-up technique.

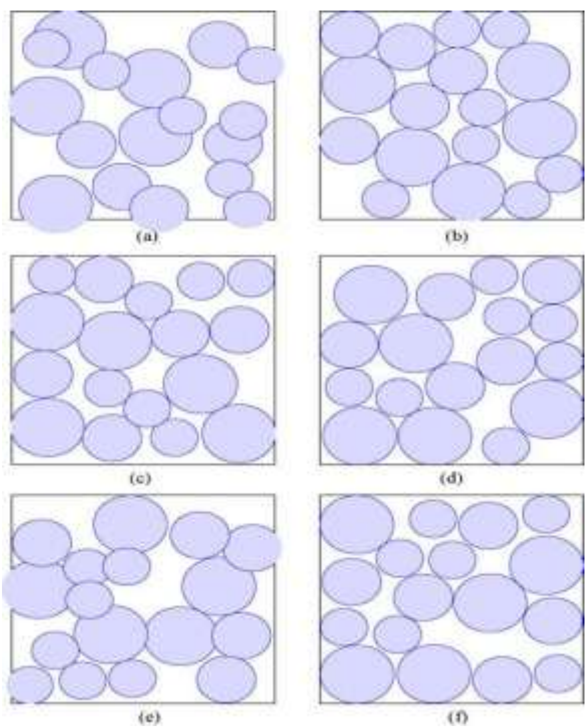

Fig. 1. Sensor deployment of instance S1-0.7 found by various methods.(a) RANDOM (Coverage: 5980.55). (b) PGA (Coverage: 6802.99). (c) MGA (Coverage: 6806.75). (d) OPTGA (Coverage: 6810.67). (e) Multi-start VFA. (Coverage: 6512.01). (f) OPTHGA (Coverage: 6814.65).

Best solutions found by RANDOM, PGA, MGA, OPTGA, VFA, and OPTHGA for the smallest instance S1-0.7, and Fig. 5 visualizes the best solutions for the largest instance S5-0.9. In both figures, we can see that the solution found by OPTHGA is the best and the solution found by RANDOM is the worst among the six methods.

Based on these results, we can conclude that MCSDP is efficiently solved by GAs, and the performance can be improved by the proposed normalization method, speed-up technique, and hybridization with local search.

Table I
Test Instances

Instance	r_1	n_1	r_2	n_2	r_3	n_3	n	α
S1-0.7	14.00	5	11.20	5	8.96	7	17	0.68
S2-0.7	12.00	6	9.60	8	7.68	10	24	0.69
S3-0.7	10.00	8	8.00	12	6.40	16	36	0.70
S4-0.7	8.00	12	6.40	18	5.12	27	57	0.70
S5-0.7	6.00	22	4.80	32	3.84	47	101	0.70
S1-0.8	14.00	5	11.20	6	8.96	10	21	0.80
S2-0.8	12.00	6	9.60	9	7.68	14	29	0.79
S3-0.8	10.00	9	8.00	13	6.40	19	41	0.79
S4-0.8	8.00	14	6.40	20	5.12	29	63	0.78
S5-0.8	6.00	25	4.80	36	3.84	55	116	0.80
S1-0.9	14.00	6	11.20	7	8.96	10	23	0.90
S2-0.9	12.00	7	9.60	11	7.68	14	32	0.89
S3-0.9	10.00	11	8.00	14	6.40	21	46	0.90
S4-0.9	8.00	16	6.40	23	5.12	34	73	0.90
S5-0.9	6.00	28	4.80	41	3.84	61	130	0.90

V. CONCLUSION

Maximum coverage sensor deployment problem (MCSDP), which frequently aroused in real-world applications. We analyzed the properties of the problem space and tried to find good sensor deployments using novel genetic algorithms. According to our analysis, the relation between the genotype space and the phenotype space of the MCSDP can be viewed in terms of quotient space. Considering this property, we devised a novel normalization method for the problem that could improve the performance of genetic algorithms. The effectiveness of the proposed normalization method and genetic algorithms were shown by extensive experiments and it can be concluded from these results that the MCSDP is efficiently solved by genetic algorithms with performance improved by our normalization method. The evaluation of sensor deployment was implemented using a Monte Carlo method that reduced time cost by a speed-up technique that started with a small number of random samples that increased with subsequent generations. Moreover the performance could be improved even further with mimetic algorithms combined with a well-designed local search.

We applied our genetic algorithms to a sensor deployment problem with static sensors, but it is expected that the proposed methodology will also show good performance with mobile sensors. Further exploration with mobile sensors is left for future study.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," IEEE Commun. Mag., vol. 40, no. 8, pp. 102–114, Aug. 2002.

[2] KAR, K. AND BANERJEE, S. 2003. Node placement for connected coverage in sensor networks. In Proceedings of WiOpt '03.

[3] XING, G., WANG, X., ZHANG, Y., LU, C., PLESS, R., AND GILL, C. 2005. Integrated coverage and connectivity configuration for energy conservation in sensor networks. Trans. Sensor Netw. 1, 1,36–72.

[4] PODURI, S. AND SUKHATME, G. S. 2004. Constrained coverage for mobile sensor networks. In Proceedings of the IEEE International Conference on Robotics and Automation '04. 165–172.

[5] MEGUERDICHIAN, S., KOUSHANFAR, F., QU, G., AND POTKONJAK, M. 2001. Exposure in wireless ad hoc sensor networks. In Proceedings of MobiCom '01. 139–150.

[6] KOUSHANFAR,F.,MEGUERDICHIAN, S., POTKONJAK, M., AND SRIVASTAVA, M. 2001. Coverage problems in wireless ad-hoc sensor networks. In Proceedings of IEEE INFOCOM '01. 1380–1387.

[7] LIU, B. AND TOWSLEY, D. 2004. A study of the coverage of large-scale sensor networks. In Proceedings of MASS '04

[8] LI, X., WAN, P., AND FRIEDER, O. 2003. Coverage in wireless ad hoc sensor networks. IEEE TransComput. 52, 6, 753–763

[9] MIORANDI, D. AND ALTMAN, E. 2005. Coverage and connectivity of ad hoc networks in presence of channel randomness. In Proceedings of IEEE INFOCOM '05. 491–502.

[10] B. Wang, "Coverage problems in sensor networks: A survey," ACM Comput. Surveys, vol. 43, no. 4, pp. 32:1–32:53, 2011.

[11] Yourim Yoon and Yong-Hyuk Kim,An Efficient Genetic Algorithm for Maximum Coverage Deployment in Wireless Sensor Networks.

[12] S.-S. Choi and B.-R. Moon, "Normalization in genetic algorithms," in Proc. Gen. Evolut. Comput. Conf. pp. 862–873, 2003.

[13] H. J. Bremermann, J. Rogson, and S. Salaff, "Global properties of evolution processes," in Natural Automata and Useful Simulations, H. H. Pattee, 2nd ed. Spartan Books, 1966, pp. 3–42.

[14] D. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning. Reading, MA, USA: Addison-Wesley, 1989.

[15] S.-S. Choi and B.-R. Moon, "Normalization in genetic algorithms," in Proc. Gen. Evolut.Comput. Conf. pp. 862–873, 2003

[16] S. Megerian, M. Potkonjak, in: J.G. Proakis (Ed.), Wireless Sensor Networks, Wiley Encyclopedia of Telecommunications, New York, 2002.

[17] K. Chakrabarty, S. S. Iyengar, H. Qi, and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," IEEE Trans. Comput., vol. 51, no. 12, pp. 1448–1453, Dec. 2002.

[18] R.-S. Chang and S.-H. Wang, "Self-deployment by density control in sensor networks," IEEE Trans. Veh. Technol., vol. 57, no. 3, pp. 1745–1755, 2008.

[19] Z. Shen, Y. Chang, H. Jiang, Y. Wang, and Z. Yan, "A generic framework for optimal mobile sensor redeployment," IEEE Trans. Veh. Technol., vol. 59, no. 8, pp. 4043–4057, Oct. 2010.

[20] G. Wang, G. Cao, T.L. Porta, and W. Zhang, "Sensor relocation in mobile sensor networks," in Proc. IEEE Infocom Conf., Mar. 2005, pp. 2302–2312.