J. Basic. Appl. Sci. Res., 3(2s)745-750, 2013

ISSN 2090-4304
Journal of Basic and Applied
Scientific Research
www.textroad.com

Sensor Placement in WSN Using the Cellular Genetic Algorithm

ShirinKhezri[1*], AkhtarHazrati[2]

[1,2]Department of Computer Engineering and Information Technology
[1]Payame Noor University, POBOX 19395-3697, Tehran, Iran
[2]Islamic Azad University, Ahar Branch, Iran

Received: June 10 2013
Accepted: July 10 2013

ABSTRACT

In this paper, we develop a robust and scalable algorithm for solving the sensor placement in distributed sensor networks for target location under constraints of the cost limitation and complete coverage. The problem is NP-complete for arbitrary sensor fields. The proposed algorithm is based on the Cellular Genetic Algorithm approach, that this algorithm is growing which combines GAs with Cellular Automata. The individuals are distributed in a grid landscape and their genetic operator is restricted to within neighborhood.

The proposed method on sensors surrounding is examined in different area and the new method performs more efficiently compared to the Simulated Annealing Algorithm and Genetic Algorithm. The experimental result indicated that CGA can improve convergence speed and maintain diversity of population.

KEYWORDS: Cellular Genetic Algorithm, Distributed Sensor Networks, Sensor Placement.

1- INTRODUCTION

In distributed sensor networks (DSNs), the sensor placement due to cost limitation is currently one of the most important research issues. When the environment is unknown, random placement is only choice and sensors may be thrown to any place by aircrafts randomly. The alternative is to deploy sensors on sensor field to guarantee a particular quality of service, if the properties of the terrain are predetermined. The field is generally divided into grids and sensors are carefully deployed at the grid points. Sensor placement strategy depends on DNS's application. If is to be used for surveillance, the placing of sensors depends on the coverage. Discrimination must be considered when they are used to solve target location problems. Sensor placement is a very challenging problem that has been proven to be NP-Hard for most of the formulations of sensor deployment. This paper focuses on grid-based placement method and we applied the Cellular Genetic Algorithm for solving this NP-Hard problem.

The results not only confirmed the successes of using new method in sensor replacement, but also they showed that the new method performs more efficiently compared to the Simulated Annealing Algorithm and Genetic Algorithm. In (Dhillon et al, 2002,2003), they present a resource-bounded optimization framework for sensor resource management under the constraints of sufficient grid coverage of the sensor field. In (Chakrabarty et al, 2002), they formulate the sensor placement problem in terms of cost minimization under coverage constraints. In (Sasikumar et al, 2002) Node placement in heterogeneous WSN is formulated using a generalized node placement optimization problem to minimize the network cost with lifetime constraint, and connectivity. In (Rajagopalan et al, 2008) they formulate and solve the sensor placement problem for efficient target localization in a sensor network, they develop a mathematical framework for the localization of the missile using multiple sensors based on Cramer-Rao Lower Bound (CRLB) analysis. In (Indu et al, 2009) they present the practical problem of optimally placing the multiple PTZ cameras to ensure maximum coverage of user defined priority areas with optimum values of parameters like pan, tilt, zoom and the locations of the cameras. Moreover in (Lin et al, 2005) a heuristic algorithm is proposed based on Simulation Annealing Algorithm to solve this problem considering the coverage and cost limitations. The rest of this work is organized as following. In section 2, we state the sensor placement problem. Section 3, is about the proposed algorithm. The performance evaluations are in section 4, and section 5 concludes the paper.

2- DEFINITION OF PROBLEM

The sensor network based on grid-based could be considered as a two or three dimensional network [4]. A set of sensors are settled on different points of grid points in order to monitor the sensor field. In this section we defined a power vector for each point of the field to show whether these sensors could cover

***Corresponding author**: Shirin Khezri, Department of Computer Engineering and Information Technology, Payame Noor University, E-mail: Sh.khezri@pnu.ac.ir

that point on the field or not, for which the number of components are as many as the number of sensors available.

Now if the Euclidean distance between each grid point and the corresponding sensor is less than the coverage radius of the sensor ($d<r$), then the coverage is assumed to be full (1), and it becomes a parallel component of that sensor on the power vector, Otherwise, the coverage is ineffective and the parallel components equaled to (0). If each point on the grid point in a sensor field can be covered by at least one sensor so that the sum of the vector components of that field equals to one, then the field is called completely covered.

In Figure *(Fig.1)*, a complete and discriminated sensor field of 4*4 with radius =1 is illustrated, in which a target can be detected at any place in the field. For example in Fig.1 the power vector for point 7 equals to (0, 1, 0, 0) which is calculated based on the sensors of 2, 8, 9 and 15. When a target appears at the grid point 7, the backend will receive reports from sensor 8.In a completely covered sensor field, when each grid point is identified by a unique power vector, the sensor field is said to be completely discriminated, as shown in Figure *(Fig.1)*. In this case, as soon as a target occurs in a grid of sensor field, it can be located by the backend according to power vector of the grid.

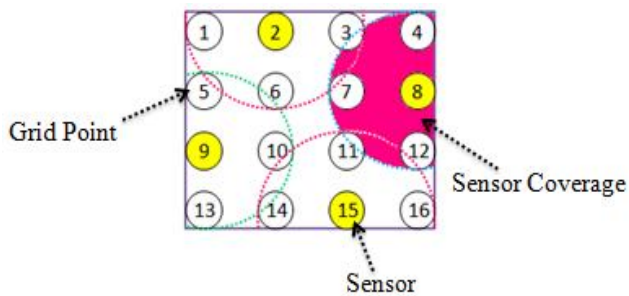

Figure.1- A complete covered and discriminated sensor field with radius =1.

3- MATHEMATICAL MODEL

The sensor placement problem is formulated herein as a combinatorial optimization problem. Complete discrimination requires that the minimum Hamming distance of the power vectors associated with any pair of grid points be at least one. High discrimination requires that the maximum distance error be minimized. The problem is, therefore, defined as a min-max model.

Given Parameters:

A= {1, 2, …, m}: Index set of the sensors' candidate locations.

B= {1, 2, …, n}: Index set of the locations in the sensor field, $m \le n$.

r_k : Detection radius of the sensor located at k, $k \in A$.

d_{ij} : Euclidean distance between location i and j, $i, j \in B$.

c_k : The cost of the sensor allocated at location k, $k \in A$.

G: Total cost limitation.

Decision Variables:

y_k : 1, if a sensor is allocated at location k and 0 otherwise, $k \in A$.

$$v_i = (v_{i1}, v_{i2}, \ldots, v_{ik}) \qquad (1)$$

v_i : The power vector of location i, where v_{ik} is 1 if the target at location i can be detected by the sensor at location k and 0 otherwise, where $i \in B$, $k \in A$.

Objective Function:

$$Z = \min_{v} \max_{(i,j)} \frac{d_{ij}}{1+K\sum_{k=1}^{m}(v_{ik}-v_{jk})^2} \qquad (2)$$

Subject to:

$$v_{ik} d_{ik} \leq y_k r_k \quad , \forall k \in A, i \in B, i \neq K \qquad (3)$$

$$\frac{d_{ik}}{r_k} > y_k - v_{ik} \quad , \forall k \in A, i \in B, i \neq K \qquad (4)$$

$$y_k = \begin{cases} 1 \text{ , if a sensor is allocated at location } k \\ 0 \text{ , otherwise} \end{cases} \qquad (5)$$

$$v_{ik} = y_k \quad , \forall k \in A, i \in B, i \neq K \qquad (6)$$

$$\sum_{k=1}^{m} c_k y_k \leq G \qquad (7)$$

$$\sum_{k=1}^{m} v_{ik} \geq 1 \quad , \forall i \in B \qquad (8)$$

When $\sum (v_{ik} - v_{jk})^2 = 0$, objective function (Z in formula (2)) introduces a penalty $d_{ij}, d_{ij} \geq 1$. As $K \to \infty$ and $\sum (v_{ik} - v_{jk})^2 > 0$, Z introduces a penalty $d_{ij} / (1+K)$ which approaches zero. Constraints (3), (4), and (6) require the relationship between sensor detection radius r_k and detection distance d_{ik}. If a target appears at grid point *i* and the grid is inside the coverage of sensor *k*, the sensor can detect the target if sensor *k* is available. Constraint (7) requires that the total deployment cost of sensors be limited by cost *G*. Constraint (8) is the complete coverage limitation. Constraint (5) is an integer constraint. *K* is an arbitrarily large number.

4- THE PROPOSED ALGORITHM

Grid applications have become a highly important area because of their potential of solving large-scale problems on science and engineering. A cellular automaton is a decentralized computing model which provides an excellent tool for complex computation. Although the cellular genetic algorithms(CGA) was initially designed for working in massive parallel machines, the model was adopted for also functioning on mono-processor machines, without any relation to parallelism at all in many researches (Whitley, 1993). The cellular genetic algorithm combines Gas with cellular automata. Individuals are placed in a regular grid of dimensions=1, 2, 3 and genetic operators are applied locally on a set made of each individual and surrounding neighborhood in order to avoid high communication overhead. So this mechanism can promote intra-neighborhood exploitation and inter-neighborhood exploration of the search space (Alba et al, 2005).

In CGAs, individuals are placed on a D-dimensional grid, each individual located in a grid (referred as a cell). The most common topology is a 2-D grid where all individuals are in "active" state. Every individual has its neighbors, and their reproduction and crossover are restricted to their nearest neighbors [see *Fig.2*]. The operator of all the cells is done synchronously.

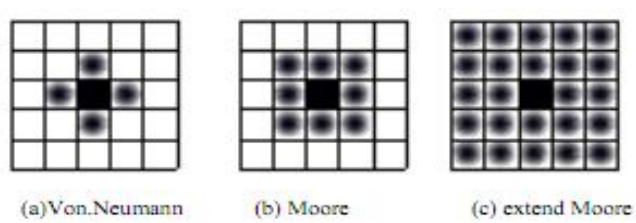

Figure.2- Structure and neighborhood of individuals

Here is a pseudo-code of CGA-SP (Dorronsoro et al 2006):
1: proc Steps Up (cga) //Algorithm parameters in 'cga-sp'

2: While not Stop Condition () do
3: for x ← 1 to WIDTH do
4: for y ← 1 to HEIGHT do
5: n list←GetNeighborhood(cga-sp,position(x,y));
6: parents←LocalSelect(n list);
7: aux indiv←Recombination(cga-sp.Pc,parents);
8: aux indiv←Mutation(cga-sp.Pm,auxindiv);
9: If coverage is satisfied based on equation (1) then
go to step11
10: Else go to step 6
11: Evaluate Fitness(aux indiv) based on equation (7)
12: Insert If Better (position(x,y),aux indiv,cga- sp,aux pop), based on Fitness Function equation (7);
13: end for
14: end for
15: cga-sp.pop←aux pop;
16: Update Statistics (cga);
17: end While
18: end proc Steps Up;

5- EXPERIMENTAL RESULTS

This section presents the computational results. First, the performance of the proposed algorithm is examined on sensor field with different area. The purpose of the experiment is to examine whether the algorithm can find the optimal solution under a minimum cost constraint.

For the experiments in this paper, we use a population including 144 individuals which distributed over 2-D grid 12×12 cells. Each individual has nine neighbors called “Moore Neighbor”. The crossover rate and mutation rate were set to 1.0 and 0.05 for all runs respectively. Computation times are expressed in term of generation. For this problem, generation is 500 and each algorithm runs 10 times repeatedand average results for different areas are calculated and compared in Table 1. These parameters are also the same in CGA and GA. These programs are run in MATLAB (v 7.6) on a personal computer (3G).

Experiments evaluate the performance of the proposed algorithm for different sensor fields. The results are compared with those obtained in Simulated Annealing and Genetic Algorithm.

First, we find a minimum sensor density for a complete covered and discriminated sensor field. Then, an attempt is made to obtain the better result by using the proposed algorithm under a sensor density constraint.

Table *(Table.1)* shows the number of sensor of three algorithms when they cover the sensor field with various area completely. In all cases, the proposed algorithm achieves the best deployment of sensor fields with a minimum sensor density based on equation (9). The required sensor density ranges between 24% and 38%. Figure (*Fig.3)* confirmed the superiority of the proposed algorithm to the SA and GA algorithms considering Sensor density (in #Sensors) vs. target area parameter. In addition, the proposed algorithm is able to trade off global search against the local search more efficiently. The proposed algorithm is, therefore, very effective and scalable.

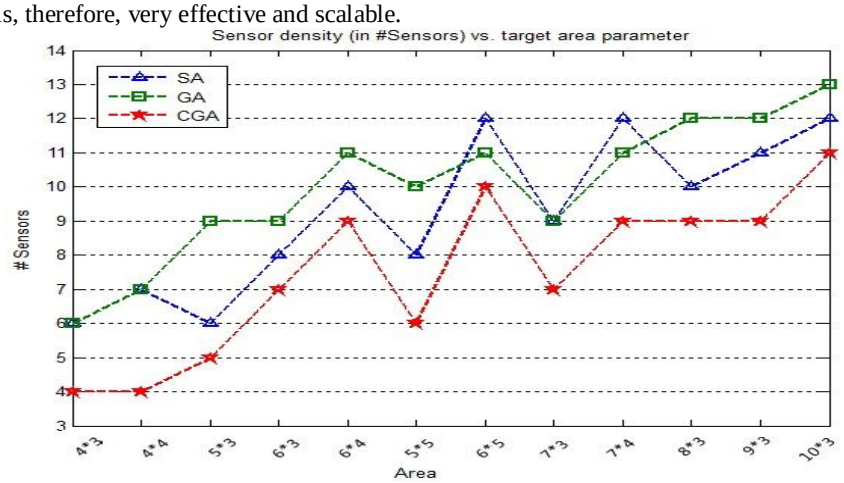

Figure.3- Sensor density (in #Sensors) vs. target area Parameter

Table1- Comparison between three algorithms and the proposed algorithm for some target area values (average of 10 runs)

	#Sensors			
Area	SA	GA	**Proposed CGA**	**Proposed CGA's Sensor Density**
4×3	6	6	4	33%
4×4	7	7	4	25%
5×3	6	9	5	33%
6×3	8	9	7	38%
6×4	10	11	9	37%
5×5	8	10	6	24%
6×5	12	11	10	33%
7×3	9	9	7	33%
7×4	12	11	9	32%
8×3	10	12	9	37%
9×3	11	12	9	33%
10×3	12	13	11	36%

Sensor density formula is:

$$Sensordensity(\%) = \left(\sum_{k=1}^{n} \frac{y_k}{n}\right) \times 100\% \qquad (9)$$

Where: $y_k = \begin{cases} 1\ , if a sensor is allocated at location k \\ 0\ , \qquad otherwise \end{cases}$

$and\ n\ is the number of grids in sensor field$

6- CONCLUSION

This paper considers the sensor placement problem for locating targets under constraints (complete coverage of sensor network with minimum costs). Firstly, we defined this NP-complete problem as a combinatorial optimization model then the CGA algorithm expanded for solving the problem. The results showed that comparing SA and GA, the proposed algorithm is able to detect more effectively the optimization solution in a limited time and costs, which provides placement of sensors to increase the coverage on the sensor field also improves the chance of escaping of local optimal. In addition the proposed algorithm is more useful, scalable and durable.

Acknowledgment

This research was supported by Payame Noor University from Iran.

REFERENCES

[1] C. Y. Chong, S. P Kumar, (2003): "*Sensor networks: evolution, opportunities, and challenges*", Proceedings of the IEEE, Vol. 91, No. 8, pp. 1247–1256.

[2] I.F. Akyildiz, W. Su, .Sankarasubramaniam, E. Cayirci, (2002): "*Wireless Sensor Networks: a survey*", Proceedings of the Computer Networks, Published by Elsevier Science, Vol. 38, No. 4, pp. 393-422.

[3] S.S. Dhillon, K**.** Chakrabarty, S.S**.** Iyengar, (2002) "*Sensor Placement for Grid Coverage under Imprecise Detections*," Proceedings of the Fifth International Conference on Information Fusion**,** , Vol. 2, No.3, pp. 1581-1587.

[4] S.S. Dhillon, K. Chakrabarty, (2003): "*Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks,"* Proceedings of the lEEE WCNC, Vol. 3, pp. 1609-1614.

[5] K. Chakrabarty, S.S**.** Iyengar, H. Qi, E. Cho, (2002): *"Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks,*" Proceedings of the IEEE Trans. on Computers, Vol. 51, No. 12.

[6] P. Sasikumar, S. K. Vasudevan, C. Vivek , V. Subashri, (2009): "*Heuristic Approaches with Probabilistic Management for Node Placement in Wireless Sensor Networks"*, Proceedings of the International Journal of Recent Trends in Engineering, Vol. 2, No. 4.

[7] R. Rajagopalan, R. Niu, Ch. K. Mohan, P. K. Varshney, A.L. Drozd, (2008): "*Sensor placement algorithms for target localization in sensor networks"*, Proceedings of the IEEE Radar Conference, pp. 1-6.

[8] S. Indu, S. Chaudhury, N. R. Mittal, A. Bhattacharyya, (2009): "*Optimal Sensor Placement for Surveillance of Large Spaces"*, Proceedings of the Third International Conference on Digital Object Identifier, pp. 1-8.

[9] F. Y. S. Lin and P. L. Chiu, (2005): "A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage/Discrimination in Sensor Networks", Proceedings of the IEEE Communication letters, Vol. 9, No. 1.

[10] D.Whitley. (1993): "Cellular Genetic Algorithm ",proc.of the 5th ICGA,Morgan Kaufmann.

[11] E. Alba and B.Dorronsoro, (2005): "The Exploration/Exploitation Tradeoff in Dynamic Cellular Genetic Algorithms", IEEE Trans. on Evolutionary Computation, Vol.9,No.2,pp.126-142.

[12] B.Dorronsoro. and E. Alba.(2006): "A Simple Cellular Genetic Algorithm for Continuous Optimization", IEEE Congress on Evolutionary Compution 2838-2844.