J. Basic. Appl. Sci. Res., 2(2)1431-1440, 2012

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

Design and Evaluation of New Intelligent Sensor Placement Algorithm to Improve Coverage Problem in Wireless Sensor Networks

Amjad Osmani

Saghez Branch, Islamic Azad University, Saghez, Iran

ABSTRACT

Adequate coverage is one of the main problems for Sensor Networks. The effectiveness of distributed wireless sensor networks highly depends on the sensor deployment scheme. Given a finite number of sensors, optimizing the sensor deployment will provide sufficient sensor coverage and save cost of sensors for locating in grid points. In many working environments, for achieving good coverage, we must able to placement of sensors in adequate places. In this article we apply the simulated annealing, genetic and learning automata as intelligent methods for solving the sensor placement in distributed sensor networks. In the distributed sensor networks, the sensor placement is a NP-complete problem for arbitrary sensor fields and it is one of the most important issues in the research fields, so the proposed algorithm is going to solve this problem by considering two factors: one is the complete coverage and the second one is the minimum cost. The proposed method is examined in different areas by C language. The results not only confirm the successes of using new method in sensor placement, also they show that the new method is more efficiently in large areas compared to other methods like PBIL.
KEY WORDS: Simulated Annealing, Genetic Algorithms, Learning Automata, Wireless Sensor Networks, Sensor Placement.

1. INTRODUCTION

WIRELESS sensor networks consist of certain amount of small and energy constrained nodes [1- 3]. A typical wireless sensor network consists of thousands of sensor nodes, deployed either randomly or according to some predefined statistical distribution, over a geographic region of interest. A sensor node by itself has severe resource constraints, such as low battery power, limited signal processing, limited computation and communication capabilities, and a small amount of memory. However, when a group of sensor nodes collaborate with each other, they can accomplish a much bigger task efficiently. One of the primary advantages of deploying a wireless sensor network is its low deployment cost and freedom from requiring a messy wired communication backbone [1, 4].

For instance, a sensor network can be deployed in a remote island for monitoring wildlife habitat and animal behavior [5-6], or near the crater of a volcano to measure temperature, pressure, and seismic activities. In many of these applications the environment can be hostile where human intervention is not possible and hence, the sensor nodes will be deployed randomly or sprinkled from air and will remain unattended for months or years without any battery replacement. Therefore, energy consumption or, in general, resource management is of critical importance to these networks. Sensor deployment strategies play a very important role in providing better QoS, which relates to the issue of how well each point in the sensing field is covered. We can address coverage problem in two main categories defined by Gage [7] in Fig. 1:

1. Blanket coverage — to achieve a static arrangement of sensor nodes which maximizes the detection rate of targets appearing in the sensing field.

2. Sweep coverage — to move a number of sensor nodes across a sensing field, such that it addresses a specified balance between maximizing the detection rate and minimizing the number of missed detections per unit area.

In this article we will focus mainly on the Blanket coverage, where the objective is to deploy sensor nodes in strategic ways such that optimal area coverage is achieved according to the needs of the underlying applications.

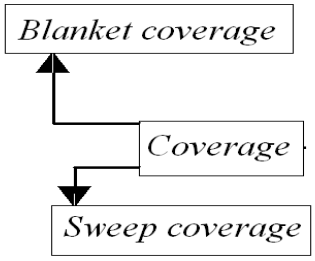


Figure.1- Coverage problem in two main categories

***Corresponding Author:** Amjad Osmani, Saghez Branch, Islamic Azad University, Saghez, Iran.
E-Mail: amjad.osmani@yahoo.com Tel.: *(+98) 914 726 5913*

2. Related Work

Several deployment strategies have been studied for achieving an optimal sensor network architecture which would minimize cost, provides high sensing coverage, be resilient to random node failures, and so on. Random placement does not guarantee full coverage because it is stochastic in nature, hence often resulting in accumulation of nodes at certain areas in the sensing field but leaving other areas deprived of nodes. Some of the deployment algorithms try to find new optimal sensor locations after an initial random placement and move the sensors to those locations, achieving maximum coverage. These algorithms are applicable to only mobile sensor networks. Research has also been conducted in mixed-sensor networks, where some of the nodes are mobile and some are static and approaches are also proposed to detect coverage holes after an initial deployment and to try to heal or eliminate those holes by moving sensors.

Zou and Chakrabarty [8] proposed an algorithm as a sensor deployment strategy to enhance the coverage after an initial random placement of sensors. It is assumed that sensors can move by "virtual force" with the force's strength determined by node distance. Cao, Wang, La Porta, and Zhang [9] considered the problem of moving some sensors from their initial random placement in order to cover some areas which were not covered by either the nature of randomness or some other effects such as wind. It is also assumed that sensors can move after gathering some information from neighbors. The algorithm proceeds in rounds. In each round, each sensor then subtracts its sensing area from its Voronoi polygon, and moves in the direction of the largest uncovered piece of area. The process repeats until no further improvement is possible. The approach appears suitable when robots, equipped with sensors, are monitoring an area, which can also be monitored by some static sensors. An alternative approach may be to use face routing [10] to estimate the size of a hole, find its centroid, estimate the number of sensors which should move toward the centroid, and provide the best possible information to sensors for their move.

Wang, Cao, and La Porta [11] proposed a proxy-based sensor deployment protocol. Instead of moving iteratively, sensors calculate their target locations based on a distributive iterative protocol. Current proxy sensors advertise the service of mobile sensors to their neighborhoods (up to certain parameter distance), searching for a better coverage location. They collect bidding messages and choose the highest bid. Then they delegate the bidder as the new proxy. Actual movement only occurs when sensors determine their final locations. If the bidding process is local, the sensor movement and the area-coverage gains may be restricted. If the bidding process includes neighbors at several hops distance, the communication overhead for bidding becomes significant. Bidding decisions are based on price (number of logical movements made so far) and distance that the moving sensors are physically supposed to move altogether. A procedure to prevent multiple healing is described, which includes some message overhead. The bidding criterion does not include lost area coverage for moving out of the current position. It is not certain whether the described procedure is always loop-free and always converging. The difference between sensing and transmission radii has a direct impact on message complexity.

[12] Proposed a scan-based movement-assisted sensor deployment method (SMART) which uses scan and dimension exchange to achieve a balanced state. In SMART, a given rectangular sensor field is first partitioned into a 2-D mesh through clustering. Each cluster corresponds to a square region and has a cluster head which is in charge of bookkeeping and communication with adjacent cluster heads. Clustering is a widely used approach in sensor networks for its support for Design simplification.

Another idea has been described in [13] for the purpose of providing location service. If the network of static sensors is disconnected, then mobile sensors will send one message to each connected component and search several perimeters.

In [14] three sensor relocation algorithms were proposed according to the mobility degree of sensor nodes. The first one, PSOA, regards the sensors in the network as a swarm and reorganizes the sensors by PSO, in the full sensor mobility case. The other two, relay shift based algorithm (RSBA) and energy–efficient fuzzy optimization algorithm (EFOA), assume relatively limited sensor mobility, i.e., the movement distance is bounded by a threshold, and to further reduce energy consumption.

In [15] for the exposure estimation the whole network is thought to be a Voronoi diagram [16] based network which is formed considering all the cluster head of the network as a single point. The clustering algorithm used here is based on Delaunay triangulated sensor nodes. The key idea of this clustering method is taken from the clustering method used for key frame-based video summarization technique [17]. Some recent work focus on sensors with limited mobility, which is motivated by the DARPA project called Intelligent Mobile Land Mine Units (IMLM) [18]. In IMLM, the mobility system is based on a hopping mechanism. Chellapan et al. [19] studied a special hopping model in which each sensor can flip (or flop) to a new location only once. In addition, the flip distance is bounded. The deployment problem is then formulated as a minimum cost, maximum-flow problem.

In [20, 21], they presented a resource-bounded optimization framework for sensor resource management under the constraints of sufficient grid coverage of the sensor field. In [22], they formulated the sensor placement problem in terms of cost minimization under coverage constraints. In [23] 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 [24] they formulated and solved the sensor placement problem for efficient target localization in a sensor network. They developed a mathematical framework for the localization of the missile using multiple sensors based on Cramer-Rao Lower Bound (CRLB) analysis. In [25] they presented 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 [26] a heuristic algorithm is proposed based on Simulation Annealing Algorithm to solve this problem considering the coverage and cost limitations.

In [27- 28], they applied Fuzzy Logic System (FLS) to re-deploy the sensors. Each individual mobile sensor uses a FLS to self-adjust its location. Therefore the deployment scheme based on FLSs is a fully distributed approach.

In [27- 28], the algorithms after random deployment of sensors at the beginning of network setup, start to redeployment of sensors to increase the coverage on the sensor field. FSPNS [27] after deciding for movement uses one of three policies depend on fuzzy rules based on neighbors density and average distance from them. FSPNS is simple and simulation shows coverage enhancement. FRED [28] uses neighbor's information about location and state to decide on movement.

In [29] they have used the Distribution Estimation Algorithms named LAEDA on sensor placement. In Learning Automata Estimation Distributed Algorithm (LAEDA), the independency of genome variables is assumed. In these algorithms a Learning Automata is used for each variable in genome. The number of actions of Learning Automata equals to number of permitted values for the corresponding variable of Learning Automata. For production of each genome sample, the Learning Automata of each variable is asked to select its own suitable action; afterwards, they give a corresponding value of selected action to the corresponding variable. Though, they can calculate the probability of a genome's production $X = (x_1,...,x_n)$ based on equation (1).

$$p(X = x) = \prod_{i=1}^{n} p(X_i = x_i) = \prod_{i=1}^{n} Grad_i^j \quad (1)$$

Where, $1 \le j \le r_i$. So $Grad_i^j$ equals to probability of action of corresponding j to value of x_i by i^{th} Learning Automata. By applying Automata in each stage, a number of N individual genomes are created, which is compatible with the number of population. Then the new population of genomes is evaluated using Evaluation Function, and Se genomes which are considered as the best genomes are chosen from this population. After applying some mechanisms which are dependent on Learning Automata Environment Model, a reinforcement signal vector is created and we apply the Learning process in each Learning Automata. Having accomplished the learning process, a new generation is produced and the above stages will be continued until a terminal condition is satisfied.

Another model of probability distribution estimation algorithms is Population Based Incremental Learning [30- 31] that is a technique which combines aspects of Genetic Algorithms and simple competitive learning. Like the GA, PBIL represents the solution set as a population set of solution vectors. In general, each solution vector in the population set, called an individual, is a possible solution of the problem. The population is produced randomly according to the probabilities specified in the probability vector. The population is evaluated and the knowledge about composing of the best individual in the population is acquired and then the probability vector is updated by pushing it towards generating good individuals in the population. After the probability vector being updated, a new generation population is produced according to the updated probability vector, and the cycle is continued until the termination condition is satisfied.

In [32] a Fuzzy Adaptive Population-Based Incremental Learning algorithm (FAPBIL) is presented based on analyzing the characteristics of traditional PBIL algorithm. Overcoming disadvantages of traditional PBIL algorithm, the proposed FAPBIL algorithm can adjust learning rate and mutation probability automatically according to the evolution degree of the algorithm's searching performed using Fuzzy Controller.

In [33- 34] they apply the modified binary particle swarm optimization algorithm for solving the sensor placement in distributed sensor networks. PSO is an inherent continuous algorithm, and the discrete PSO proposed to be adapted to discrete binary space.

We can summarize the above methods in Fig. 2.

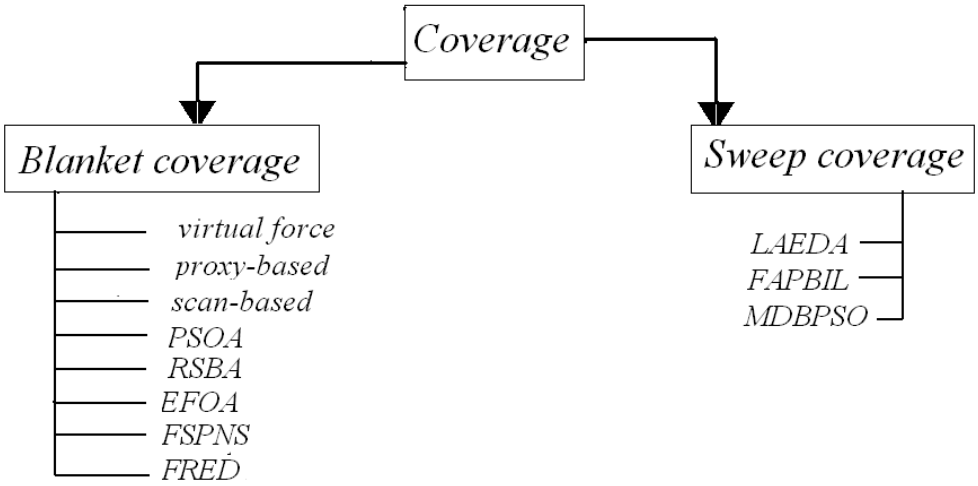


Figure.2- Organizing placement methods in two main categories

The rest of this work is organized as following. In section 3, 4 we describe the sensor placement problem. Section 5 is about the proposed algorithm. The performance evaluations are in Section 6 and Section 7 concludes this article.

3. Definition of Problem

A grid-based sensor field can be represented as a collection of two- or three-dimensional grid points [20]. A set of sensors can be deployed on the grid points to monitor the sensor field. In this article, we consider the detection model of a sensor to be a 0/1 coverage model. The coverage is assumed to be full (1) if the distance between the grid point and the sensor is less than the detection radius of the sensor (rd). Otherwise, the coverage is assumed to be ineffective (0). If any grid point in a sensor field can be detected by at least one sensor, we call the field is completely covered, as shown in Fig. 3. A power vector is defined for each grid point to indicate whether sensors can cover a grid point in a field. As shown in Fig. 3, the power vector of grid point 1 is (1, 0, 0, 0) corresponding to sensor 2, 8, 9 and 15. 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 Fig. 3. Sometimes, due to some resource limitations, a completely discriminated sensor field cannot be constructed. Consequently, these may lead to wrong determinations, whenever a target occurs at any one of the indistinguishable grid points. Positioning accuracy, therefore, becomes a major consideration in solving the problem. Distance error is one of the most natural criteria to measure positioning accuracy. The distance error of two indistinguishable grid points is defined as the Euclidean distance between them.

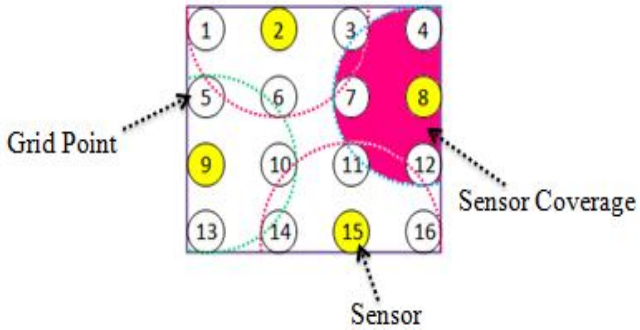


Figure.3- A complete covered and discriminated sensor field with radius =1

4. 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, $k \in A$.

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

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

Subject to:

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

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

$$v_{ik} = y_k \quad ,\forall k \in A, i \in B, i \neq K \qquad (5)$$

$$\sum_{k=1}^{m} c_k y_k \leq G \qquad (6)$$

$$\sum_{k=1}^{m} v_{ik} \geq 1 \quad ,\forall i \in B \qquad (7)$$

$$v_{ik}, y_k = 0 or 1 \quad ,\forall k \in A, i \in B \qquad (8)$$

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 (5) 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 (6) requires that the total deployment cost of sensors be limited by cost G. Constraint (7) is the complete coverage limitation. Constraint (8) is an integer constraint. K is an arbitrarily large number.

5. Suggested Algorithm

Herein we describe a new algorithm for better placement of sensors in sensor field area. SAGLA (Simulated Annealing + Genetic + Learning Automata) uses Simulated Annealing, Genetic and Learning Automata. The optimized placement of sensors in this article is how some grid points are chosen to hold the sensors. Our algorithm represents the relationship between the potential answers and the chromosomes. So the chromosomes of the potential answers $\Phi_t (t = 1,2,...,T)$ can be represented to binary bits, that is $\Phi_t \{s_1, s_2, s_3 ..., s_m\}, s_k \in \{0,1\}$ Where m is the number of all candidate grid points, sk is the decision variable which indicates whether a sensor is allocated to the candidate grid point k (sk = 1 means allocating, sk = 0 means not).

Each chromosome is generated based on a probability vector. We use a probability vector in background for each chromosome. As you can see in Fig. 4, probability vector maps to sensor grid state and number 1 means a sensor is allocated to a grid point and vice versa.

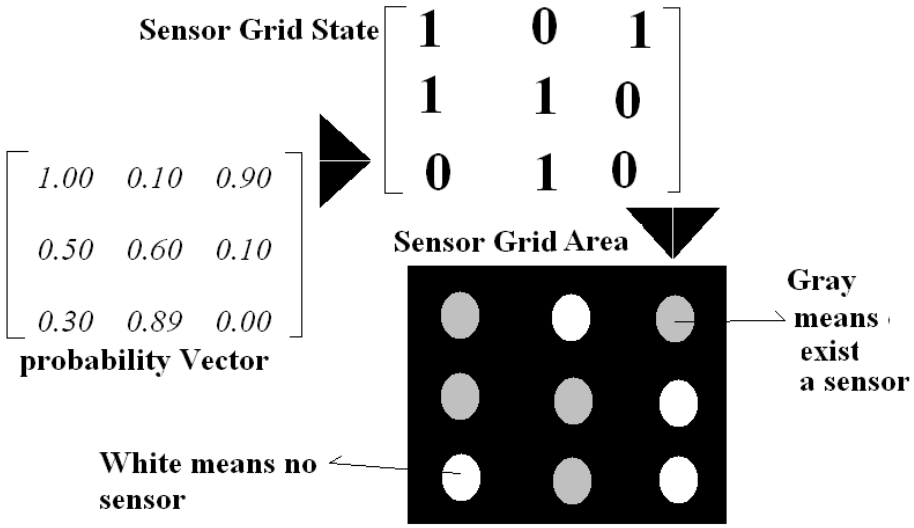


Figure.4- An example for mapping probability vector to sensor grid area

In SAGLA, the SA (simulated annealing) algorithm is characterized by a rule for randomly generating a new solution in the neighborhood of the current solution. Our used operators to reach the neighborhood solution are usual genetic operators and learning automata. The new solution is accepted if min $\{1, e^{-\Delta/T}\} \geq p$, where p is a uniform random number in range of (0, 1), and $\Delta = f(\mathbf{s}n) - f(\mathbf{s}c)$ is the difference result of objective values of the new solution $\mathbf{s}n$ and the current solution $\mathbf{s}c$. Herein we use energy as E for objective value (f). Note that if Δ is negative, then min $\{1, e^{-\Delta/T}\} = 1$, which means that

new solution always is accepted. A certain number of iterations *L* are performed at fixed temperature *T* and the temperature is reduced according to a cooling schedule.
The steps of SAGLA algorithm are:

1) The algorithm generates a population based on probability vector of 0.5. Each chromosome is generated based on a probability vector. The bits (as genes) in each binary string (as chromosome) are selected randomly and set to 1 or 0. It means probability vector is set to 0.5 for each gene of each chromosome. Note that each chromosome can map to a state for sensor grid and it must satisfy constraint 6.
2) The algorithm computes energy of chromosomes using formula (9).
3) The algorithm after generating *n* (random) solutions obtains the energy average of these *n* chromosomes and assigns it to Δ, which is ultimately used in determining the value of *T*0. The idea is to set a value to *T*0 such that the probability of accepting an average bad move early in the algorithm is being equal to *p*0. Algebraically, *p*0 = $e^{-\Delta/T0}$ which implies that $T0 = \Delta/\ln(p0)$.
4) The algorithm generates a chromosome based on probability vector of 0.5.
   The chromosome can map to a state for sensor grid and it must satisfy constraint 6.
5) The algorithm computes energy of current chromosome using function of formula (9) as Ec (Energy of current chromosome).

$$E = \max_{\forall(i,j)} \frac{d_{ij}}{1+K\sum_{k=1}^{m}(v_{ik}-v_{jk})^2} \qquad (9)$$

6) The algorithm selects the state of first chromosome as best state and its energy value as Ebest (Energy of best chromosome).
   The algorithm uses a genetic operator like mutation for reach to next probability vector for generating next chromosome as neighborhood state. In mutation operation our algorithm selects genes with the probability of pm and exchanges previous gene with next gene and sets previous gene to 1-p as you can see in Fig. 6. Note that each chromosome can map to a state for sensor grid and it must satisfy constraint 6.

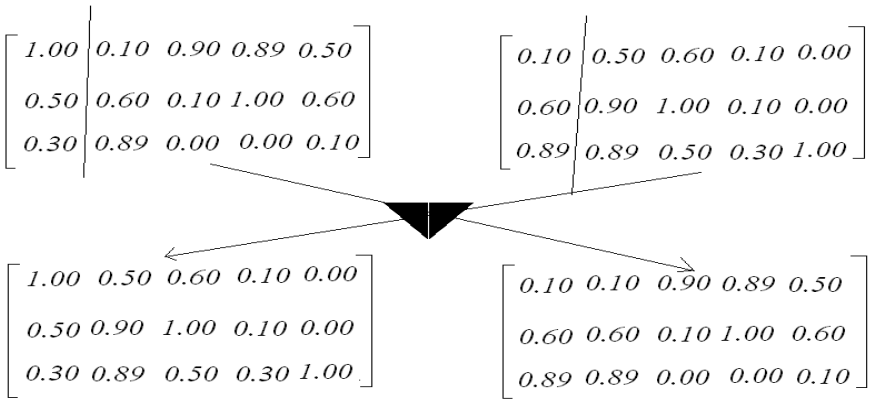


Figure.5- An example for performing crossover operation

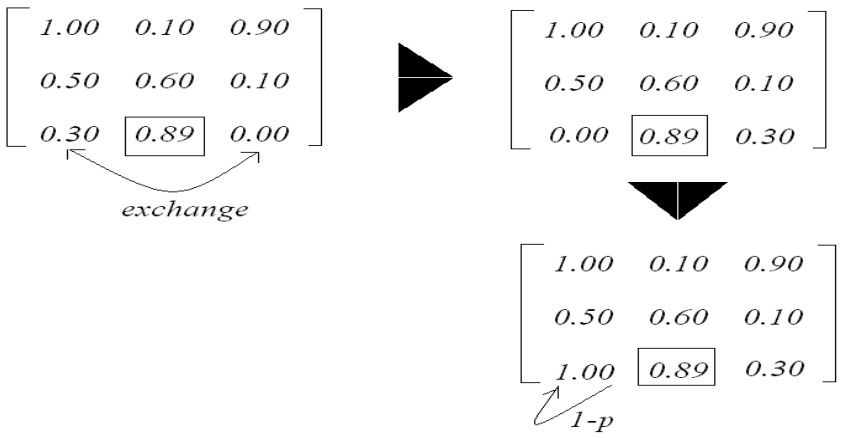


Figure.6- An example for performing mutation operation

7) The algorithm computes energy of current chromosome using function of formula (9) as En (Energy of next chromosome).
8) The algorithm computes ΔE of En and Ec.
9) The algorithm accepts new state with the probability of min {1, $e^{-\Delta E/Tk}$}.
10) After acceptance of new state, the algorithm set Ec=En.
11) After acceptance of new state, the algorithm saves new state in a temporal memory with e entries.
12) After acceptance of new state, the algorithm selects the state of next chromosome as best state and its value as Ebest if its value is less than Ebest.
13) The algorithm rejects new state with the probability of (1-min {1, $e^{-\Delta E/Tk}$}).
14) After rejection, the algorithm performs crossover operation. This operation uses two parents to produce two childes. One of parents is current chromosome and the other is chromosome of best state. Two childes are evaluated and the one with minimum energy is selected**.** The algorithm performs crossover operation like Fig. 5.
15) Let k=k+1, Tk=Tk-1 * α.
16) The algorithm is finished if the number of generated chromosomes reaches to a desired value.
17) The algorithm uses learning automata (formula (10)) for updating probability vector for next round.

$$Pi(n+1)=\begin{cases} Pi(n)+\theta\beta(n)(1-Pi(n)) & (10.1) \\ Pi(n)-\theta\beta(n)(Pi(n)) & (10.2) \end{cases} \quad (10)$$

Where θ is learning rate and β(n) is computed using formula (11).

$$\beta(n)=\frac{Ec(n)-\min(E)}{\max(E)-\min(E)} \quad (11)$$

Where min(E) and max(E) are minimum and maximum energy values of best values up to now. Ec(n) is energy value of current chromosome in nth round. P(n) means probability vector of new chromosome in nth round. It is computed for each gene in the chromosome. So it is updated based on formula (10. 1) if gene jth of chromosome is 1 otherwise formula (10. 2) is used.

18) The algorithm uses usual genetic operators like crossover and mutation for reach a probability vector for generating next chromosome as neighborhood state. The chromosome can map to a state for sensor grid and it must satisfy constraint 6. The algorithm can use two types crossover. First is single-parent and second is with two parents. Herein we use first method. In first type the algorithm selects genes with the probability of pc and exchange genes over a cycle with radius of one gene in clockwise mode like Fig. 7.

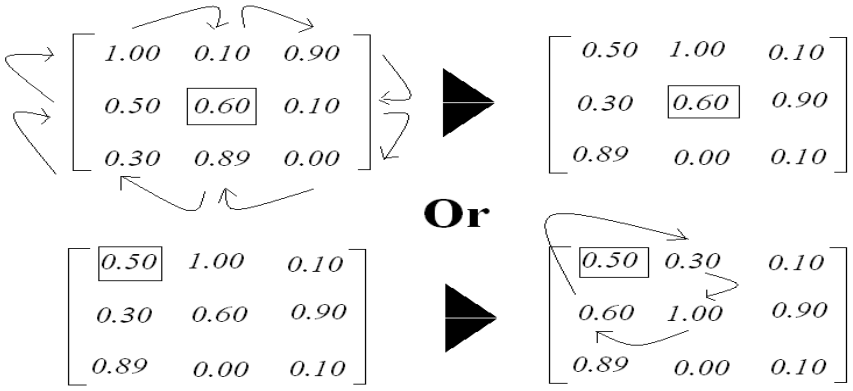


Figure.7- Two examples for performing crossover operation

Additionally the algorithm uses mutation operator. In mutation operation our algorithm selects genes with the probability of pm and exchanges previous gene with next gene and sets previous gene with 1-p as you can see in Fig. 6. The algorithm uses new probability vector for generating next chromosome.

19) The algorithm checks temporal memory to find an entry and as soon as finding an entry the algorithm goes to 17 otherwise it goes to 8. Input/ output to/ from temporal memory are like a fifo (first in first out) queue structure. This is an advantage and can help us to find a state that has been reached by algorithm before and the algorithm avoids from happening that state in future for at least e round. e is the number of entries in temporal memory.

6. Experimental Section

This section presents the computational results. Firstly the performance of the proposed algorithm is evaluated when small sensor fields are deployed. The purpose of this experiment is to examine whether the algorithm can find the optimal solution in small sensor fields. Then, the performance results in the case of larger sensor fields are presented.

As all sensors have the same deployment cost, the cost constraint, Constraint (6), can be expressed as a limit on the number of sensors. The cost of each sensor is equal to one. Also the radius of each sensor is one. In MDPSO we assume that the population size is 30, $c_1 = c_2 = 2$ and $v_{max} = 6$. The value of w is considered between 0.2 and 0.9. The parameters of FAPBIL, PBIL and LAEDA are set as Table 1. In the table, Pop-Size means population size in each generation, Pm means mutation probability, LR means learning rate and Se is selection size for next generation across previous generation. In LAEDA and PBIL algorithms, a high value of *Se* genomes was chosen for updating the genome`s probability model. In all experiments, we assume the value of *Se* as a value equals to half of population of each generation and Learning Rate is 0.01. In SA algorithm, the parameters of the cooling schedule are $\alpha=0.75$ and $\beta=1.3$. The initial value of r and t are $5n$ and 0.1 and n is the number of grids in the sensor field. The frozen temperature, t_f, is $t_0/30$. In SAGLA algorithm, the parameter of the cooling schedule is $\alpha=0.55$. θ is equal to 0.41, pc=0.4 and pm=0.3 (pc means crossover probability and pm is mutation probability) and k is 1000. Detection radius of each sensor (rd) is 1. Parameter of e in SAGLA is equal to 3.

TABLE1. THE PARAMETERS OF FAPBIL, PBIL AND LAEDA

Parameters	Pop-Size	Pm	LR	Se
LAEDA	50	-	0.01	Pop/2
PBIL	50	0.2	0.01	Pop/2
FAPBIL	50	*Fuzzy Adaptive*	*Fuzzy Adaptive*	-

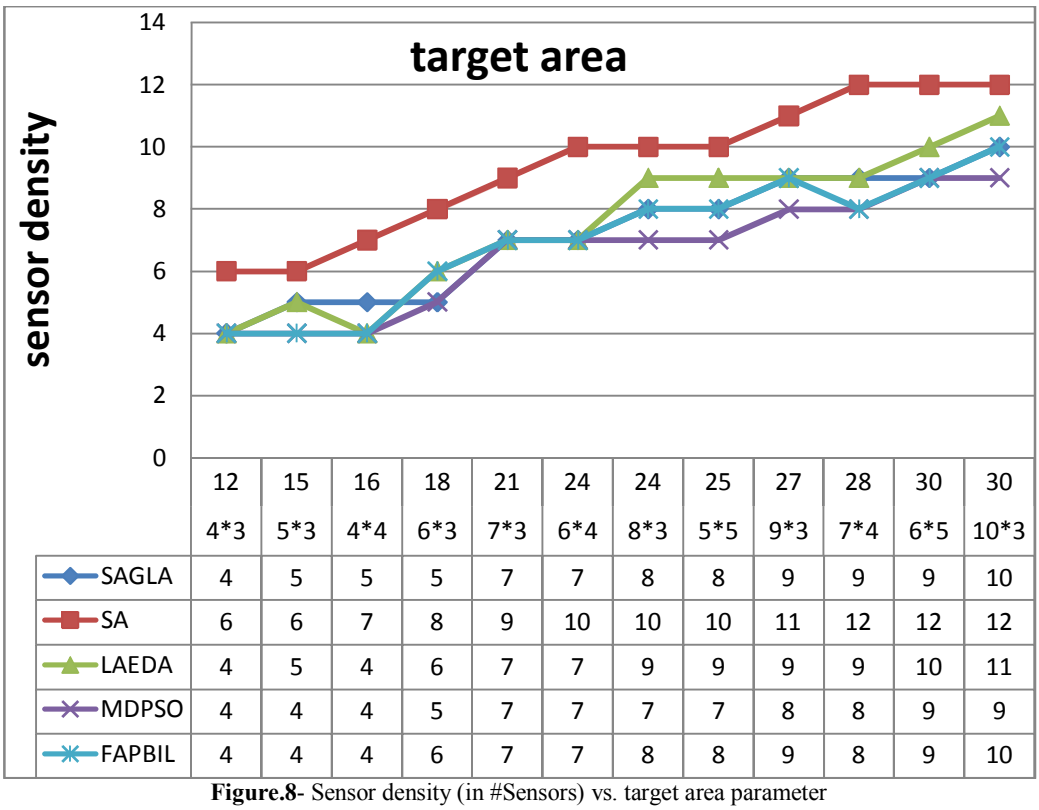
Each algorithm is run for 20 times and average results for different areas are calculated and compared in. The above methods are examined in a benchmark environment that has provided by *C language*.

A. Experiment I

Experiment I, evaluates the performance of our proposed algorithms for smaller rectangular sensor fields which have no more than 30 grid points. The results are compared with those obtained in SA, MDPSO, FAPBIL and LAEDA. Each algorithm is run 20 times and the average results for different areas are calculated and compared in Fig. 8 that confirms the superiority of the proposed algorithm against the SA, MDPSO, FAPBIL and LAEDA algorithms considering Sensor density (in #Sensors) vs. target area parameter.

B. Experiment II

In this experiment, a large sensor field, with 15 × 15 grid points is considered. The radius of the each sensor is one. Obtained results using the proposed algorithm are compared with the best solution obtained by other approaches and Fig. 9 is showing that.



	12 4*3	15 5*3	16 4*4	18 6*3	21 7*3	24 6*4	24 8*3	25 5*5	27 9*3	28 7*4	30 6*5	30 10*3
SAGLA	4	5	5	5	7	7	8	8	9	9	9	10
SA	6	6	7	8	9	10	10	10	11	12	12	12
LAEDA	4	5	4	6	7	7	9	9	9	9	10	11
MDPSO	4	4	4	5	7	7	7	7	8	8	9	9
FAPBIL	4	4	4	6	7	7	8	8	9	8	9	10

Figure.8- Sensor density (in #Sensors) vs. target area parameter

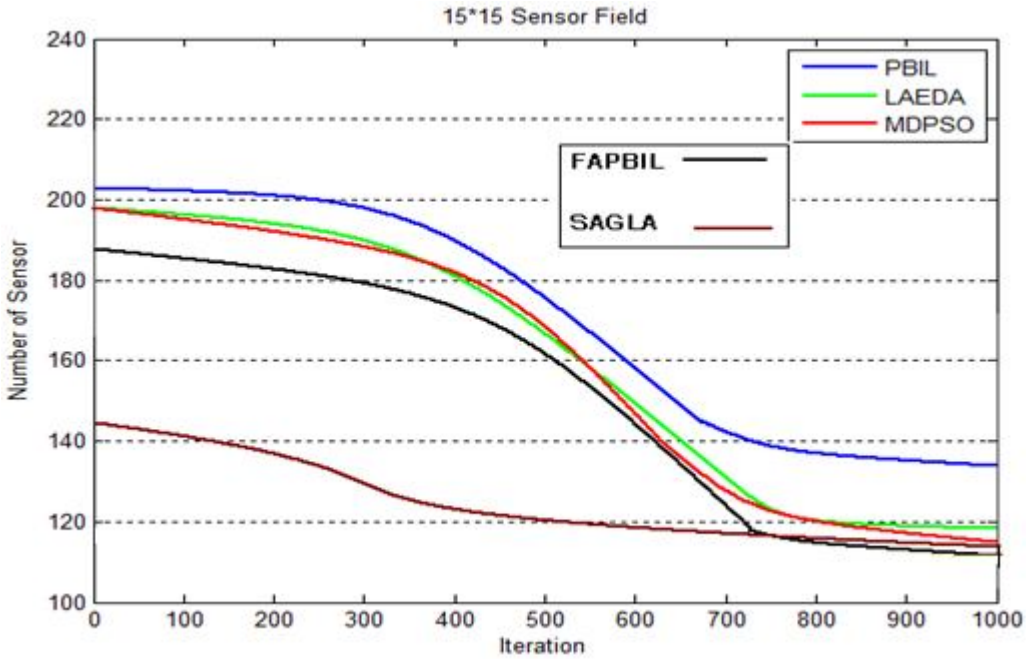


Figure.9- Sensor density (in #Sensors) for 15*15 sensor field

7. Conclusion

This article considers the sensor placement problem for locating targets under constraints (complete coverage of sensor network with minimum cost). Firstly, we defined this N-complete problem as a combinatorial optimization model then the method SAGLA for solving the problem. The results show that our algorithm compared to simulated annealing algorithm, are more able to detect the optimization solution, which provides placement of sensors to increase the coverage on the sensor field. Since sensor placement in the Wireless Sensor Networks (WSN) is important, we should find better intelligent algorithms.

Acknowledgement

This research was supported by Islamic Azad University of Saghez.

REFERENCES

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

2. Pottie, G.J. (1998). *Wireless sensor networks* , Proceedings of Information Theory Workshop, (pp.139–140).

3. Pottie, G.J., Caiser, W. (2000). *Wireless sensor networks*, Proceedings of the Commun .ACM, (Vol 43,No.5).

4. Akyildiz, I., Su, W., Sankarasubramaniam, Y., Cayirci, E. (2002). *Wireless sensor networks A survey,* Proceedings of the Comput . Networks, (Vol 38, No.2, pp.393–422).

5. Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J. (2002). *Wireless sensor networks for habit at monitoring*, Proceedings of the 1st ACMInt.Workshop on Wireless Sensor Networks and Applications (WSNA'02), (pp.88–97).

6. Qi, H., Iyengar, S. S., Chakrabarty, K. (2001). *Distributed sensor fusion – a review of recent research*, Proceedings of the Journal of the Franklin Institute, (Vol. 338, pp. 655–668).

7. Gage, D.W. (1992). *Command control for many-robot systems,* Proceedings of the 19th Annual AUVS Technical Symp. Reprinted in Unmanned Syst.Mag. (Vol 10, No.4, pp.28–34).

8. Zou., Y., Chakrabarty, K. (2004). Sensor deployment and target localization in distributed Sensor networks, Proceedings of the Trans .IEEE Embedded Comput .Syst. (Vol 3, No.1).

9. Wang, G., Cao, G., La Porta, T. (2006). *Movement-assisted sensor deployment*, Proceedings of the IEEE Transactions On Mobile Computing, (Vol.5, No.6).

10. Bose, P., Morin, P., Stojmenovic, I., Urrutia, J. (2001). *Routing with guaranteed delivery in Adhoc wirelesss networks*, Proceedings of the Wireless Networks, (Vol 7, No.6).

11. Wang, G., Cao, G., La Porta, T. (2004). *Proxy-based sensor deployment for mobile sensor networks,* Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'04).

12. Yang, Sh., Li, M., Wu, J. (2007). *Scan-Based Movement-Assisted Sensor Deployment Methods in Wireless Sensor Networks*, Proceedings of the IEEE Transactions On Parallel And Distributed Systems, (Vol. 18, No. 8).

13. Stojmenovic, I. (1999). *A Scalable Quorum Based Location Update Scheme for Routing In Ad Hoc Wireless Network",* Technical Report TR-99-09, SITE, University of Ottawa.

14. WU, X., CHO, J., AURIOL, B. J.D', LEE, S. (2007).*Mobility-Assisted Relocation for Self-Deployment in Wireless Sensor Networks,* Proceedings of the IEICE TRANS.COMMUN., (Vol E90–B, NO.8).

15. Rahman, Md. O., Razzaque, Md. A., Hong, Ch. S. (2007). *Probabilistic Sensor Deployment in Wireless Sensor Network: A New Approach,* Proceedings of the ICACT, (pp. 12-14).

16. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M. B. (2001). *Coverage Problems in Wireless Ad-hoc SensorNetworks*, Proceedings of the International Conference on Mobile Computing and Networking.

17. Padmavathi, M., Yong, R., Yelena, Y.*(2006). Key frame-based Video Summarization using Delanunay Clustering*, Proceedings of the International Journal on Digital Libraries, (Vol 6, pp. 219 – 232).

18. .(2007). http://www.darpa.mil/ato/progarms/shm/index.html.

19. Chellappan, S., Bai, X., Mam B., Xuan, D. (2005). *Mobility Limited Flip-Based Sensor Networks Deployment* Proceedings of the IEEE Mobile Ad Hoc and Sensor Systems (MASS).

20. Dhillon, S.S., Chakrabarty, K., Iyengar, S.S. (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).

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

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

23. Sasikumar, P., Vasudevan, S. K., Vivek, C. , Subashri, V. (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).

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

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

26. Lin, F. Y. S., Chiu, P. L. (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).

27. Osmani, A., Haghighat, A. T., Dehghan, M., Emdadi, P. (2010). *FSPNS: Fuzzy Sensor Placement Based on Neighbors State,* Proceedings of the International Conference on Computer Modeling and Simulation, (IEEE) UKSIM,.

28. Osmani, A., Dehghan, M., Pourakbar, H., Emdadi, P. (2009). *Fuzzy-Based Movement-Assisted Sensor Deployment Method in Wireless Sensor Networks*, Proceedings of the International Conference on Computational Intelligence, Communication Systems and Networks, (IEEE) CICSYN.

29. Khezri, S., Osmani, A., Gholami, M. (2011). *Estimation of Distribution Algorithm Based on Learning Automata for Sensors Placement in Wireless Sensor Networks*, Proceedings of the 3nd National Conference on Computer/Electrical and IT Engineering (CEIC).

30. Baluja, S., (1994). *Population Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning*, Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, Pennsylvania.

31. Baluja, S., Caruana, R., (1995). *Removing The Genetics from The Standard Genetic Algorithm*, Proceedings of ICML, (Morgan Kaufmann Publishers),(pp.38-46).

32. Khezri, S., Meybodi, M., Osmani, A. (2011). *Fuzzy Adaptive PBIL based Sensor Placement in WSN*, Proceedings of the International Symposium on Computer Networks and Distributed Systems, (IEEE)CNDS.

33. Khezri, S., Faez, K., Osmani, A. (2011). *Modified Discrete Binary PSO Based Sensor Placement in WSN*, Proceedings of the International Conference on Computational Intelligence and Communication Networks, (IEEE) CICN.

34. Khezri, S., Faez, K., Osmani, A. (2011). *An Intelligent Sensor Placement Method to Reach a High Coverage in Wireless Sensor Networks*, International journal of grid and high performance computing (IJGHPC), (IGI publication), (Vol.3 No.3).