

Swarm Based Sensor Deployment Optimization in Ad hoc Sensor Networks

Wu Xiaoling, Shu Lei, Yang Jie, Xu Hui, Jinsung Cho, and Sungyoung Lee

Department of Computer Engineering, Kyung Hee University, Korea
{xiaoling, sl8132, yangjie, xuhui, sylee}@oslab.khu.ac.kr
chojs@khu.ac.kr

Abstract. In ad hoc sensor networks, sensor nodes have very limited energy resources, thus energy consuming operations such as data collection, transmission and reception must be kept at a minimum. This paper applies particle swarm optimization (PSO) approach to optimize the coverage in ad hoc sensor networks deployment and to reduce cost by clustering method based on a well-known energy model. Sensor nodes are assumed to be mobile, and during the coverage optimization process, they move to form a uniformly distributed topology according to the execution of algorithm at base station. The simulation results show that PSO algorithm has faster convergence rate than genetic algorithm based method while demonstrating good performance¹.

1 Introduction

Recent military operations have limitations of surveillance missions performed by high-altitude platforms (UAV, U2, satellite) even when equipped with state of the art sensors. Most of the limitations are inherent to long-distance surveillance and cannot be resolved by any improvement in the onboard-sensor technology [1].

In order to get a clear understanding of the situation on the ground, it is important to observe from close range, using remote sensing device placed in the region of interest (ROI) to form a sensor network. Ad hoc sensor networks that employ ad hoc networking have become an area of intense research activity. In most cases, a large number of wireless sensor devices can be deployed in hostile areas without human involved, e.g. by air-dropping from an aircraft for remote monitoring and surveillance purposes. Such airdropped networks are called *ad hoc* sensor networks to distinguish them from other types of sensor networks where nodes are laid out in some fixed predetermined pattern. Due to their attractive characteristics, ad hoc sensor networks have been applied to many military and civil applications such as target tracking, surveillance, and environmental control. Usually, once the sensors are deployed on the ground, their data are transmitted back to the base station to provide the necessary situational information.

¹ Dr. Sungyoung Lee is the corresponding author.

The limited energy storage and memory of the deployed sensors prevent them from relaying data directly to the base station. It is therefore necessary to form a cluster based topology, and the cluster heads (CHs) provide the transmission relay to base station such as a satellite. And the aircraft carrying the sensors has a limited payload, so it is impossible to randomly drop thousands of sensors over the ROI, hoping the communication connectivity would arise by chance; thus, the mission must be performed with a fixed maximum number of sensors. In addition, the airdrop deployment may introduce uncertainty in the final sensor positions. Though many scenarios adopt random deployment for practical reasons such as deployment cost and time, random deployment may not provide a uniform sensor distribution over the ROI, which is considered to be a desirable distribution in sensor networks. These limitations motivate the establishment of a planning system that optimizes the sensor reorganization process after initial random airdrop deployment assuming sensor node mobility, which results in the maximum possible utilization of the available sensors.

There exist a lot of research work [2], [3], [4] related to the placement of sensor nodes in network topology design. Most of them focused on optimizing the location of the sensors in order to maximize their collective coverage. However only a single objective was considered in most of the research papers, other considerations such as energy consumption minimization are also of vital practical importance in the choice of the network deployment. Self-deployment methods using mobile nodes [4, 9] have been proposed to enhance network coverage and to extend the system lifetime via configuration of uniformly distributed node topologies from random node distributions. In [4], the authors present the virtual force algorithm (VFA) as a new approach for sensor deployment to improve the sensor field coverage after an initial random placement of sensor nodes. The cluster head executes the VFA algorithm to find new locations for sensors to enhance the overall coverage. They also considered unavoidable uncertainty existing in the precomputed sensor node locations. This uncertainty-aware deployment algorithm provides high coverage with a minimum number of sensor nodes. However they assumed that global information regarding other nodes is available. In [1], the authors examined the optimization of wireless sensor network layouts using a multi-objective genetic algorithm (GA) in which two competing objectives are considered, total sensor coverage and the lifetime of the network. However the computation of this method is not inexpensive.

In this paper, we attempt to solve the coverage problem while considering energy efficiency using particle swarm optimization (PSO) algorithm, which can lead to computational faster convergence than genetic algorithm used to solve the deployment optimization problem in [1]. Sensor nodes are assumed to have mobility, and during the coverage optimization process, they move to form a uniformly distributed topology according to the execution of algorithm at the base station. To the best of our knowledge, this is the first paper to solve deployment optimization problem by PSO algorithm.

In the next section, the PSO algorithm is introduced and compared with GA. Modeling of sensor network and the deployment algorithm is presented in section 3, followed by simulation results in section 4. Some concluding remarks and future work are provided in section 5.

2 Particle Swarm Optimization

PSO, originally proposed by Eberhart and Kennedy [5] in 1995, and inspired by social behavior of bird flocking, has come to be widely used as a problem solving method in engineering and computer science.

The individuals, called, particles, are flown through the multidimensional search space with each particle representing a possible solution to the multidimensional problem. All of particles have fitness values, which are evaluated by the fitness function to be optimized, and have velocities, which direct the flying of the particles. PSO is initialized with a group of random solutions and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" factors. The first one, called *pbest*, is the best fitness it has achieved so far and it is also stored in memory. Another "best" value obtained so far by any particle in the population, is a global best and called *gbest*. When a particle takes part of the population as its topological neighbors, the best value is a local best and is called *lbest*. After each iteration, the *pbest* and *gbest* (or *lbest*) are updated if a more dominating solution is found by the particle and population, respectively.

The PSO formulae define each particle in the D-dimensional space as $X_i = (x_{i1}, x_{i2}, x_{i3}, \dots, x_{iD})$ where i represents the particle number, and d is the dimension. The memory of the previous best position is represented as $P_i = (p_{i1}, p_{i2}, p_{i3}, \dots, p_{iD})$, and a velocity along each dimension as $V_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{iD})$. The updating equation [6] is as follows,

$$v_{id} = \varpi \times v_{id} + c_1 \times rand() \times (p_{id} - x_{id}) + c_2 \times rand() \times (p_{gd} - x_{id}) \quad (1)$$

$$x_{id} = x_{id} + v_{id} \quad (2)$$

where ϖ is the inertia weight, and c_1 and c_2 are acceleration coefficients.

The role of the inertia weight ϖ is considered to be crucial for the PSO' s convergence. The inertia weight is employed to control the impact of the previous history of velocities on the current velocity of each particle. Thus, the parameter ϖ regulates the trade-off between global and local exploration ability of the swarm. A large inertia weight facilitates global exploration, while a small one tends to facilitate local exploration, i.e. fine-tuning the current search area. A suitable value for the inertia weight ϖ balances the global and local exploration ability and, consequently, reduces the number of iterations required to locate the optimum solution. Generally, it is better to initially set the inertia to a large value, in order to make better global exploration of the search space, and gradually decrease it to get more refined solutions. Thus, a time-decreasing inertia weight value is used. The initial swarm can be generated randomly [7].

PSO shares many similarities with GA. Both algorithms start with a group of a randomly generated population, have fitness values to evaluate the population, update the population and search for the optimum with random techniques. However, PSO does not have genetic operators like crossover and mutation. Particles update them-

selves with the internal velocity. They also have memory, which is important to the algorithm [8].

Compared with GA, PSO is easy to implement, has few parameters to adjust, and requires only primitive mathematical operators, computationally inexpensive in terms of both memory requirements and speed while comprehensible. It usually results in faster convergence rates than GA. This feature suggests that PSO is a potential algorithm to optimize deployment in a sensor network.

3 The Proposed Algorithm

First of all, we present the model of wireless sensor network. We assume that each node knows its position in the problem space, all sensor members in a cluster are homogeneous and cluster heads are more powerful than sensor members. Sensing coverage and communication coverage of each node are assumed to have a circular shape without any irregularity. The design variables are 2D coordinates of the sensor nodes, $\{(x_1, y_1), (x_2, y_2), \dots\}$. And the sensor nodes are assumed to be mobile. Many research efforts into the sensor deployment problem in wireless sensor network [4, 9] make this sensor mobility assumption reasonable.

3.1 Optimization of Coverage

We consider coverage as the first optimization objective. It is one of the measurement criteria of QOS of a sensor network.

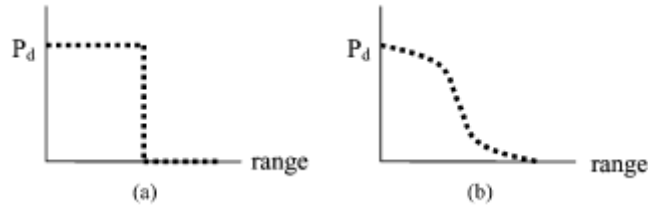


Fig. 1. Sensor coverage models (a) Binary sensor and (b) stochastic sensor models

The coverage of each sensor can be defined either by a binary sensor model or a stochastic sensor model as shown in Fig. 1 [9]. In the binary sensor model, the detection probability of the event of interest is 1 within the sensing range, otherwise, the probability is 0. In the stochastic sensor model, the probability of detection of the event of interest follows a decaying function of distance from the sensor. In this paper, the binary sensor model is employed and coverage is defined as the ratio of the union of areas covered by each node and the area of the entire ROI, as shown in Eq (3). Here, the covered area of each node is defined as the circular area within its sensing radius [9].

$$C = \frac{\sum_{i=1, \dots, N} A_i}{A} \quad (3)$$

where

- A_i is the area covered by the i^{th} node;
- N is the total number of nodes;
- A stands for the area of the ROI.

In order to prevent recalculating the overlapped area, the coverage here is calculated using Monte Carlo method by meshing the network space, i.e., by creating a uniform grid in the ROI. All the grid points being located in the sensing area are labeled 1 otherwise 0, depending on whether the Euclidean distance between each grid point and the sensor node is longer or shorter than sensing radius, as shown in Fig 2. Then the coverage can be approximated by the ratio of the summation of ones to the total number of the grid points.

If a node is located well inside the ROI, its complete coverage area will lie within the ROI. In this case, the full area of that circle is included in the covered region. If a node is located near the boundary of the ROI, then only the part of the ROI covered by that node is included in the computation.

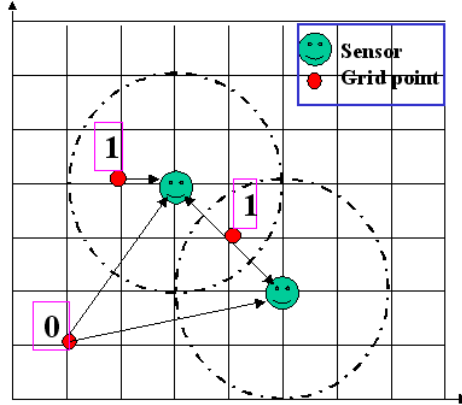


Fig. 2. Sensing coverage calculation (dashed circle indicating the sensing area boundary)

3.2 Optimization of Energy Consumption

After optimization of coverage, all the deployed sensor nodes move to their own positions. Now we can disregard the assumption of sensor mobility since our goal is to minimize energy usage in a cluster based sensor network topology by finding the optimal cluster head (CH) positions. For this purpose, we assume a power consumption model [10] for the radio hardware energy dissipation where the transmitter dissipates energy to run the radio electronics and the power amplifier, and the receiver dissipates energy to run the radio electronics. This is one of the most widely used

models in sensor network simulation analysis. For our approach, both the free space ($distance^2$ power loss) and the multi-path fading ($distance^4$ power loss) channel models were used. Assume that the sensor nodes inside a cluster have short distance dis to cluster head but each cluster head has long distance Dis to the base station. Thus for each sensor node inside a cluster, to transmit an l -bit message a distance dis to cluster head, the radio expends

$$E_{TS}(l, dis) = lE_{elec} + l\epsilon_{fs}dis^2 \quad (4)$$

For cluster head, however, to transmit an l -bit message a distance Dis to base station, the radio expends

$$E_{TH}(l, Dis) = lE_{elec} + l\epsilon_{mp}Dis^4 \quad (5)$$

In both cases, to receive the message, the radio expends:

$$E_R(l) = lE_{elec} \quad (6)$$

The electronics energy, E_{elec} , depends on factors such as the digital coding, modulation, filtering, and spreading of the signal, here we set as $E_{elec}=50nJ/bit$, whereas the amplifier constant, is taken as $\epsilon_{fs}=10pJ/bit/m^2$, $\epsilon_{mp}=0.0013pJ/bit/m^2$.

So the energy loss of a sensor member in a cluster is

$$E_s(l, dis) = l(100 + 0.01dis^2) \quad (7)$$

The energy loss of a CH is

$$E_{CH}(l, Dis) = l(100 + 1.3 \times 10^{-6} \times Dis^4) \quad (8)$$

Since the energy consumption for computation is much less than that for communication, we neglect computation energy consumption here.

Assume m clusters with n_j sensor members in the j^{th} cluster C_j . The total energy loss E_{total} is the summation of the energy used by all sensor members and all the m cluster heads:

$$E_{total} = l \sum_{j=1}^m \sum_{i=1}^{n_j} \left(100 + 0.01dis_{ij}^2 + \frac{100}{n_j} + \frac{1.3 \times 10^{-6} Dis_j^4}{n_j} \right) \quad (9)$$

Because only 2 terms are related to distance, we can just set the fitness function as:

$$f = \sum_{j=1}^m \sum_{i=1}^{n_j} \left(0.01dis_{ij}^2 + \frac{1.3 \times 10^{-6} Dis_j^4}{n_j} \right) \quad (10)$$

4 Performance evaluation

The PSO starts with a “swarm” of sensors randomly generated. As shown in Fig. 3 is a randomly deployed sensor network with coverage value 0.4484 calculated using Eq. (3). A linear decreasing inertia weight value from 0.95 to 0.4 is used, decided according to [6]. Acceleration coefficients c_1 and c_2 both are set to 2 as proposed in [6]. For optimizing coverage, we have used 20 particles, which are denoted by all sensor nodes coordinates, for our experiment in a 50×50 square sensor network, and the maximum number of generations we are running is 500. The maximum velocity of the particle is set to be 50. The sensing range of each sensor is set to be 5 units. An upper bound on the coverage is given by the ratio of the sum of the circle areas (corresponding to sensors) to the total area of the sensor field. In this simulation, the upper bound evaluates to be 0.628, which is calculated from the perfect uniform distribution case without any overlapped area. The coverage is calculated as a fitness value in each generation.

After optimizing the coverage, all sensors move to their final locations. Now the coordinates of potential cluster heads are set as particles in this static sensor network. The communication range of each sensor node is 15 units with a fixed remote base station at (25, 80). We start with a minimum number of clusters acceptable in the problem space to be 4. The node, which will become a cluster head will not have any restriction on the transmission range. The nodes are organized into clusters by the base station. Each particle will have a fitness value, which will be evaluated by the fitness function (10) in each generation. Our purpose is to find the optimal location of cluster heads. Once the position of the cluster head is identified, if there is no node in that position then a potential cluster head nearest to the cluster head location will become a cluster head.

We also optimized the placement of cluster head in the 2-D space using GA. We used a simple GA algorithm with single-point crossover and selection based on a roulette-wheel process. The coordinates of the cluster head are the chromosomes in the population. For our experiment we are using 10 chromosomes in the population. The maximum number of generations allowed is 500. In each evolution we update the number of nodes included in the clusters. The criterion to find the best solution is that the total fitness value should be minimal.

Fig. 4 is the coverage optimization results after 6 runs. Compared with the upper bound 0.628, the difference between them is small. Fig. 5 shows the convergence rate of PSO and GA. We ran the algorithm for both approaches several times and in every run PSO converges faster than GA, which was used in [1] for coverage and lifetime optimization. The main reason for the fast convergence of PSO is due to the velocity factor of the particle.

Fig. 6 shows the final cluster topology in the sensor network space after coverage and energy consumption optimization when the number of clusters in the sensor space is 4. We can see from the figure that nodes are uniformly distributed among the clusters compared with the random deployment as shown in Fig 3. The four stars denote cluster heads, the small circles are sensor members, and the dashed circles are communication range of sensor nodes. The energy saved is the difference between the

initial fitness value and the final minimized fitness value. In this experiment, it is approximately 16.

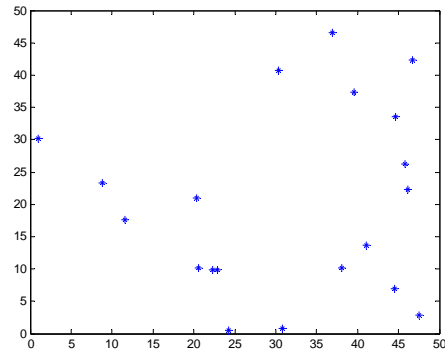


Fig. 3. Randomly deployed sensor network (Coverage value=0.4484)

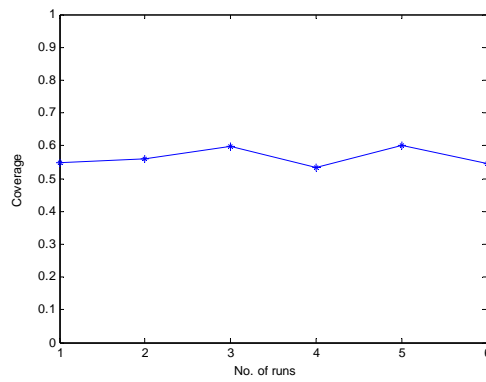


Fig. 4. Optimal coverage results for 6 runs

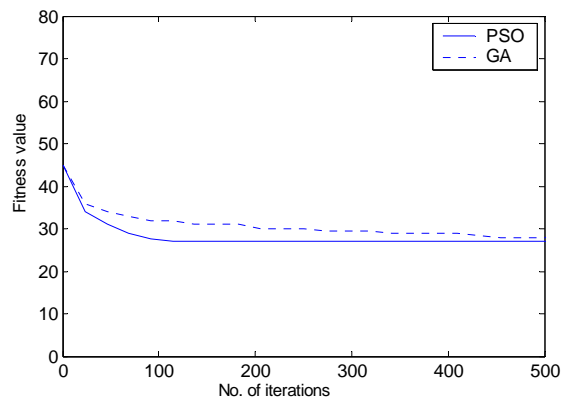


Fig. 5. Comparison of convergence rate between PSO and GA based on Eq. (10)

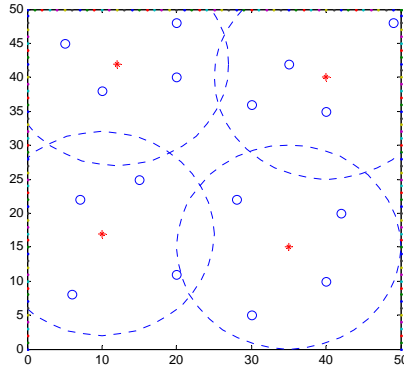


Fig. 6. Energy efficient cluster formation using PSO

5 Conclusions and Future Work

The application of PSO algorithm to optimize the coverage in ad hoc sensor network deployment and energy consumption in cluster-based topology is discussed. We have used coverage as the first optimization objective to place the sensors with mobility, and a distance based energy model to reduce cost based on clustering method. The simulation results show that PSO algorithm has faster convergence rate than GA based layout optimization method while demonstrating good performance.

In the future work, we will take the uncertainty in the position of the sensors due to the initial random deployment into account. Moreover, other objectives, such as time and distance for sensor moving will be further studied.

Acknowledgement

This work was supported by grant No. R01-2005-000-10267-0 from Korea Science and Engineering Foundation in Ministry of Science and Technology.

References

1. Damien B. Jourdan, Olivier L. de Weck: Layout optimization for a wireless sensor network using a multi-objective genetic algorithm. IEEE 59th Vehicular Technology Conference (VTC 2004-Spring), Vol.5 (2004) 2466-2470

2. K. Chakrabarty, S. S. Iyengar, H. Qi and E. Cho: Grid coverage for surveillance and target location in distributed sensor networks. *IEEE transactions on computers*, Vol.51 (2002) 1448-1453
3. A. Howard, M.J. Mataric and G. S. Sukhatme: Mobile sensor network deployment using potential fields: a distributed, scalable solution to the area coverage problem. *Proc. Int. Conf. on distributed Autonomous Robotic Systems* (2002) 299-308
4. Y. Zou and K. Chakrabarty: Sensor deployment and target localization based on virtual forces. *Proc. IEEE Infocom Conference*, Vol. 2 (2003) 1293-1303
5. Kennedy and R. C. Eberhart: Particle Swarm Optimization. *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia (1995) 1942-1948
6. Yuhui Shi, Russell C. Eberhart: Empirical study of Particle Swarm Optimization. *Proceedings of the 1999 Congress on Evolutionary Computation*, Vol. 3 (1999) 1948-1950
7. K.E. Parsopoulos, M.N. Vrahatis. Particle Swarm Optimization Method in Multiobjective Problems. *Proceedings of the 2002 ACM symposium on Applied computing*, Madrid, Spain (2002): 603- 607
8. <http://www.swarmintelligence.org/tutorials.php>
9. Nojeong Heo and Pramod K. Varshney: Energy-Efficient Deployment of Intelligent Mobile Sensor Networks. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans*, Vol. 35, No. 1 (2005): 78 - 92
10. Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. *IEEE Transactions on Wireless Communications*, Vol. 1, No. 4 (2002): 660 - 670