Application of Multiobjective Particle Swarm Optimization to maximize Coverage and Lifetime of wireless Sensor Network

[1]Deepak Kumar Chaudhary, [1]Professor Rajeshwar Lal Dua
[1] M.Tech Scholar, [2] Professors, HOD,
Department of Electronics & Communication Engineering, JNU, Jaipur, India

Abstract

The autonomous nodes used for monitoring environments are known as Wireless Sensor Networks. The major use of Multi-Objective particle swarm optimization for maximizing coverage & life time using wireless sensor networks is described. Wireless sensor networks are being tremendous popular due to increasing number of applications for these networks. This paper introduces a proposal to extend heuristic called "Particle Swarm Optimization "(PSO) to deal with Multi-objective optimization problems.

This paper describes a new approach for energy efficient layout of wireless sensor network in which sensors communicate with each other to transmit their data to a high energy communication node which acts as an interface between data processing unit and sensors. Optimization of sensor locations is essential to provide communication for a longer duration. It discusses an energy efficient layout with good coverage based on Multi-objective Particle Swarm Optimization algorithm.

Keywords: Heuristic, Multi-objective Optimization (MOO), Particle Swarm Optimization (PSO), Wireless Sensor Network (WSN).

I. Introduction

The use and development of heurioptististic based multiobjective optimization a technique has significantly grown in last few years. Particle swarm optimization (PSO) is a reactively recent heuristic inspired by the choreography of a bird flack. Despite its current success in diverse optimization tasks, PSO remains as one of the heuristics for which not much work on multiobjective optimization has been done so far.

The main purpose of the sensor placement problem is to determine optimal locations of sensors for maximizing the information collected from the sensor network. But equally important are factors such as the energy consumption of the network. This results in a tradeoff between multiple objectives such as coverage and energy consumption.

In this paper, we address the optimization of the multiple objectives described above by adopting a multi-objective optimization framework. The approach is very simple to implement, it is population based, it uses an external memory called "repository" and a geographically-based approach to maintain diversity-MOPSO.

II. Particle Swarm Optimization

A. The original version

The particle swarm concept originated as a simulation of a simplified social system. The original intent was to graphically simulate the graceful but unpredictable choreography of a bird flock. These methods were proposed by Kennedy and Eberhart (1995). At some point in the evolution of the algorithm, it was realized that the conceptual model was, in fact, an optimizer.

PSO is similar to a genetic algorithm (GA) in that the system is initialized with a population of random solutions. Each particle keeps track of its coordinates in the problem space which are associated with the best solution, it has achieved so far. This fitness value is called pbest. Another "best" value that is tracked by the global version of the particle swarm optimizer is the overall best value, and its location, obtained so far by any particle in the population. This location is called gbest. The particle swarm optimization concept consists of, at each time, step, changing the velocity (accelerating) each particle toward its pbest & gbest locations (global version of PSO).

B. Implementation of global version of PSO

The (original) process for implementing the global version of PSO is as follows:

1. Initialize a population (array) of particles with random position & velocities (on d dimensions in the space).
2. For each particle, evaluate the desired optimization fitness function in d variables.
3. Compare particle's fitness evaluation with particle's pbest. If current value is better than pbest, the set pbest value equal to the current value, and the pbest location equal to the current location in d-dimensional space.

4. Compare fitness evaluation with the population's overall previous best. If current value is better than gbest, then reset gbest to the current particles array index and value.
5. Change the velocity and position of the particle according to equation (1) & (2) respectively.

$$V_{id} = V_{id} + C1{*}rand\,(\,)\,{*}\,(P_{id} - X_{id}) + C2\,{*}\,Rand\,(\,)\,{*}\,(P_{gd} - X_{id}) \quad (1)$$

$$X_{id} = X_{id} + V_{id} \quad (2)$$

6. Loop to step (2) until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations.

Particles velocities on each dimension are clamped to maximum velocity V_{max} . If the sum of accelerations would cause the velocity on that dimension to exceed V_{max}, then the velocity on that dimension is limited to Vmax.

If V_{max} is too high, particles might fly past good solutions. If V_{max} is too small, particles may not explore sufficiently beyond locally good regions. The acceleration constants C1 & C2 in equation (1) represent the weighting of the stochastic acceleration terms that pull each particle toward pbest and gbest positions.

III. Problem formulation

A. Modeling of Wireless Sensor Network

It is assumed that each node knows its position in the search space and all sensor nodes are homogeneous. High energy communication node (HECN) is assumed to be more powerful than sensor nodes. A flat square surface is considered in which HECN is placed at the center for convenience. The sensing area of each node is assumed to have a circular shape with radius R_{sens}. The communication range of each node is defined by the area of a circle with radius R_{comm} .Initially nodes are assumed to have equal energy. It is assumed that for every data transmission, the energy decreases by one unit. The co-ordinates of the sensor nodes (x_1, y_1), (x_2, y_2). . . . . . are considered as design variables. Sensor nodes are assumed to have certain mobility.

B. Objectives of MOPSO

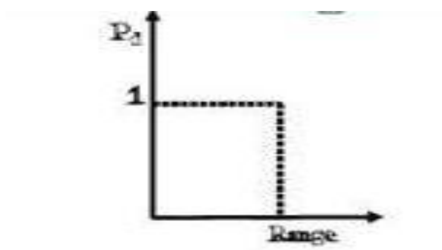
Two major objectives of MOPSO are considered.

• Maximize the total coverage of the sensor network: f1

• Maximize the lifetime of the sensor network: f2

Coverage

It is one of the measurements of quality of service (QoS) of a sensor network. The coverage of each sensor can be defined either by a binary sensor model is shown in Figure .1 or a stochastic sensor model is shown in Figure 2. In the binary sensor model shown in Figure 1, the detection probability of the event of interest is one within the sensing range; otherwise, the probability is zero. Coverage of a network using binary sensor model is determined by finding the union of sensing areas defined by location of each sensor and Rsens. Although the binary sensor model is simpler, it does not take the uncertainty factor in sensor measurement into consideration. The binary sensor model is given by



$$C_{ij}(x,y) = \begin{cases} 1 & \text{for } d_{ij}(x,y) \le R_{sens} \\ 0 & \text{for } d_{ij}(x,y) > R_{sens} \end{cases} \quad \textbf{(3)}$$

Figure 1: A binary sensor coverage model

The sensor field is represented by an mxn grid. $d_{ij}(x,y)$ denotes the Euclidean distance between a sensor node at (x, y) and any grid point at (i, j). The distances are measured in units of grid points. Equation 3 expresses the coverage $C_{ij}(x,y)$ of a grid point at (i, j) by a sensor at (x, y). The coverage for the entire grid is calculated as the fraction of grid points covered. In reality, sensor measurements are imprecise; hence the coverage needs to be expressed in probabilistic terms. In the stochastic sensor model shown in Figure 2, the probability of detection follows an exponential decaying function of distance from the sensor. The stochastic sensor model is given in Equation 4.

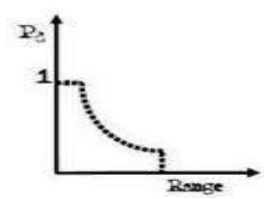


Fig 1: A binary sensor

$$C_{ij}(x,y) = \begin{cases} 1 & \text{for} \quad d_{ij}(x,y) \le (R_{sens} - R_e) \\ e^{(-\lambda\alpha)^{\beta}} & \text{for} \quad (R_{sens} + R_e) < d_{ij}(x,y) < R_{sens} \\ 0 & \text{for} \quad d_{ij}(x,y) \le (R_{sens} - R_e) \end{cases} \quad (4)$$

R_e ($R_e < R_{sens}$) is a measure of the uncertainty in sensor measurement, a=$d_{ij}(x,y) - (R_{sens} - R_e)$, and λ and β are parameters that measure the detection probability when there is an uncertainty in sensor detection. The coverage for the entire sensor field is calculated as the fraction of grid points that exceeds the threshold C_{th}. So the first objective is maximization of coverage. This objective can be calculated by the following expression:

$$\textbf{Max Coverage (f1)} = \frac{\bigcup_{i=1,\ldots\ldots,N}}{A} A_i \quad (5)$$

where A_i is the area covered by the ith node, N is the total number of nodes and A is the area of the region of interest.

Lifetime
The second objective considered is maximization of lifetime. Lifetime is defined as the time until one of the participating nodes run out of energy. This objective can be calculated by the subsequent expression:

$$\textbf{Max Lifetime (f2)} = \frac{T_{failure}}{T_{max}} \quad (6)$$

where $T_{failure}$ is the maximum number of sensing cycles before failure of any node and Tmax is the maximum number of possible sensing cycles. In every sensing cycle, the data from every node is routed to HECN through a route of minimum weight. Dijkstra algorithm is used to find out the route with minimum weight. These two objectives are competing with each other. The coverage objective will try to spread out the nodes for maximizing coverage while resulting in high energy loss and small lifetime. The lifetime objective will try to arrange the nodes as close as possible to the HECN for reducing loss in energy which results in poor coverage.

III. Proposed Multi-objective particle swarm optimization

A. Description of the proposed approach
A Pareto ranking scheme could be the straightforward way to extend the approach to handle multiobjective optimization problems. We implemented a mechanism such that each particle may choose a different guide. Our mechanism is based on the generation of hypercube which are produced dividing the search space explored.
In order to construct a direct relationship between the problem domain and the PSO particles for this problem, every particle represents coordinates of N number of nodes. So each particle represents a network layout. The proposed MOPSO algorithm is composed of the following steps:

1. Initialize the population pop:
   (a) For i=0 to Max /*Max=number of particles*/
   (b) Initialize pop[i]
2. Initialize the speed of each particle:
   (a) For i=0 to Max
   (b) VEL[i] = 0
3. Evaluate each of the particles in pop.
4. Store the positions of the particles that represent nondominated vectors in the repository REP.
5. Generate hypercubes of the search space and locate the particles using these hypercubes as coordinate system.
6. Initialize the memory of each particle. This memory is also stored in the repository.
   (a) For i=0 to MAX (b) PBESTS[i] = pop[i]
7. WHILE maximum number of cycles has not been reached DO
   (a) Compute the speed of each particle using the following expression.
      VEL[i] = W×VEL[i] + R1 × (PBESTS[i]-pop[i]) + R2× (REP[h]-pop[i])
      Where W (inertia weight) takes a value of 0.4. R1 & R2 are random numbers in the range (0 to 1); PBESTS[i] is the best position that the particle i has had; REP[h] is a value that is taken from the repository; the index h is

selected in the following way: those hypercubes containing more than one particle are assigned a fitness equal to the result of dividing any number x>1 (we used x=10 in our experiments) by the number of particles that they contain. Pop[i] is the current value of particle i.

(b) Compute the new position of the particles adding the speed produced from various step:

$$POP[i] = POP[i] + VEL[i] \qquad \text{(i)}$$

(c) Maintain the particles within the search space in case they go beyond its boundaries.

(d)Evaluate each of the particles in POP.

(e) Update the contents of REP together with the geographical representation of particles within the hypercubes.

(f) When the current position of the particle is better than the position contained in its memory, the particle's . position is updated using:

$$PBEST[i] = POP[i] \qquad \text{(ii)}$$

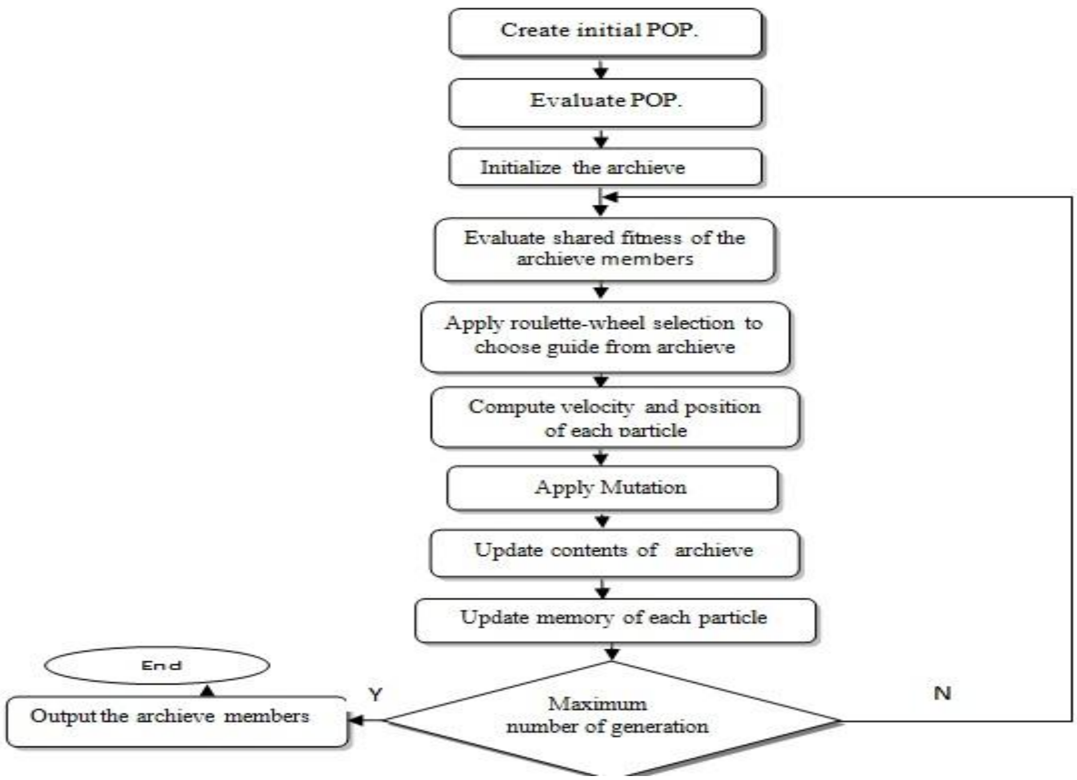
The criterion to decide what position from memory should be retained is simply to apply Pareto.

(g) Increment the loop counter.

(h) END WHILE

B. Flowchart of Multi Objective Particle Swarm Optimization

MOPSO that used in this study is dominance-based method that was proposed by Coello et al. In this algorithm the best nondominated solutions have ever been visited are stored in a memory space calling archieve. The main objective of every multi-objective optimization algorithm is to find Pareto-optimal set. Traditional type of assigning fitness function is aggregation–based method, where the fitness function is a weighted sum of the objective functions. However this classic approach can be very sensitive to precise aggregation of goals and tend to be ineffective and inefficient. The flowchart of this algorithm is shown in Fig. 3.



V. Results

The MOPSO algorithm starts with a "swarm" of particles randomly generated where each particle represents a network layout represented by sensor co-ordinates. The coverage and lifetime of the particles are then calculated. The archive containing non-dominated Pareto optimal set of solutions is developed according to the Pareto optimal dominance developed by Coello and Lechuge.

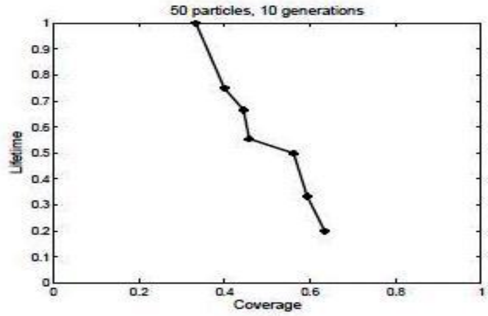


Figure 4: Pareto front for a WSN with 10 sensors

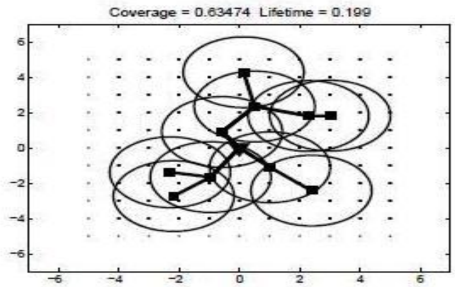


Figure 5: Pareto-optimal layout with best coverage for WSN with 10 sensors, 50 particles,10 generations

Hence sensors acting as communication relay loss more energy. The layout shown in Figure 6 is example of another pareto optimal layout available to the user. It is more interesting to look at the Pareto fronts obtained using two different sensor models as shown in Figure 7.

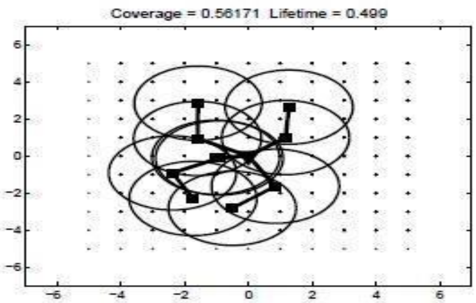


Figure 6: Example of another pareto-optimal layout for a WSN with 10 sensors, 50 particles, 10 generations

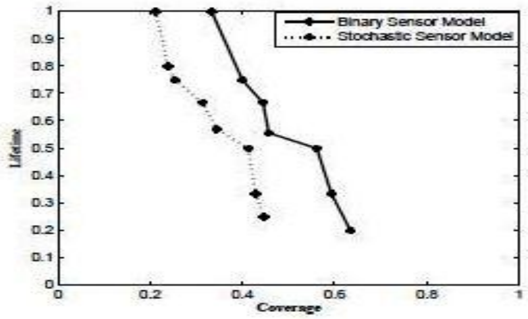


Figure 7: Pareto fronts obtained with different sensor models for a WSN with 10 sensors, 50 particles, 10 generations

The input parameters taken for simulation are as follows: grid size = 10*10, number of nodes = 10, number of particles = 50, number of generations = 10, R_{sens}= 2, R_{comm} = 2, Re = 1, λ = 0.5, β = 0.5, C_{th} = 0.7. Finally a well-populated Pareto front is obtained, from the external archive, which gives a solution set of layouts for optimization. The Pareto front obtained for a binary sensor model is shown in Figure 4. Two Pareto optimal layouts are shown in Figure 5 and 6 to illustrate the variety of layouts available. The effect of different sensor models on the Pareto front is shown in Figure 7. The improvement in Pareto front with increase in number of generations of MOPSO algorithm is also shown in Figure 8. The layout shown in Figure 5 is the layout with best coverage. For getting more coverage the particles spread out to minimize the overlapping region. Many sensors transmit their own data as well as act as communication relay for other far away sensors.

VI. Conclusion

In this paper, an energy efficient layout with good coverage for a WSN is considered. The application of MOPSO to maximize coverage and lifetime simultaneously is discussed. Thus, the aim of the proposed algorithm is to locate good non-dominated solutions under time pressure. In this paper we have also considered the deployment problem for mobile wireless sensor networks. It is important to indicate that PSO is an unconstrained search technique. Therefore, it is necessary to develop an additional mechanism to deal with constrained multiobjective optimization problems. For this purpose, we have proposed a multi objective approach for the deployment of nodes to improve upon an irregular initial deployment of nodes. Coverage and lifetime are taken as the two conflicting objectives for achieving a set of layouts. Depending on the application, the user can choose a layout from the set of solutions. The performance of the MOPSO algorithm is determined by the computation time and uniformity of the solutions on the Pareto front. Simulation results show that the MOPSO algorithm obtains a better set of solutions as compared to single objective algorithms and other deployment algorithms.

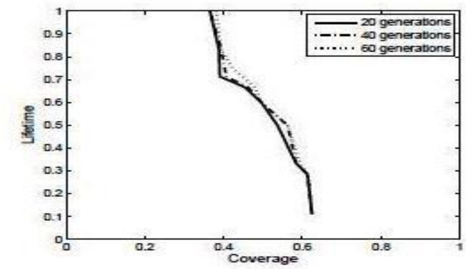


Figure 8: Pareto fronts obtained for a WSN with 10 sensors and 50 particles

VII. Future Work

More general data distribution, and also more sophisticated co-operative schemes with each node cooperating with a subset of nearby nodes, are useful extensions and will be studied in future work. We studied the LMS implementation operating with Gaussian signals. Other strategies can be studied using the formulation presented here, such as the distributed normalized LMS (dNLMS), the distributed affine projection algorithms (dAPA) and distributed RLS implementations. If both the desired and the input data are corrupted by impulsive noise, then the Huber Prior Error Feedback-Leat Square Lattice (H-PEF-LSL) algorithm gives a good performance. In practice, a WSN is divided into multiple sub-regions for easy layout, organization and management. In future work, we will also take energy consumption due to sensor movement into account.

References

[1] M. Rabbat and R. Nowak, “Distributed optimization in sensor networks,” in Pro-ceedings of the 3rd International Symposium on Information processing in sensor networks, April 2004, pp. 20–27.

[2] Kennedy, J. and R. Eberhart, “Particle swarm optimization,” *Proceedings of IEEE International Conference on Neural Network*, Piacataway, NJ, 1995.

[3] Engelbrecht, A. P., *Fundamentals of Computational Swarm Intelligence*, John Wiley & Sons, 2005.

[4] Coello, C. A. C., G. T. Pulido, and M. S. Lechuga, “Handling multiple objectives with particle swarm optimization,” *IEEE Trans. Evolutionary Computat.*, Vol. 8, 256–279, 2004.

[5] S. Haykin, Adaptive filter theory. Englewood Cliffs, NJ: Prentice-Hall, 2001.

[6] S. Dhillon, K. Chakrabarty, and S. Iyengar, “Sensor placement for grid coverage under imprecise detections,” in Proceedings of Fifth International Conference on Information Fusion, vol. 2, Seol, Korea, 2002, pp. 1581–1587.

[7] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Transactions on Evolutionary Computation, 3(4):257-271, 1999.

VII. Author Biographies

A. RAJESHWOR LAL DUA

Head Deptt. Of Electronics & Communication Engineering

Professor Rajeshwar Lal Dua a Fellow Life Member of IETE and also a Life member of I.V.S & I.P.A, former "Scientist F" is the most well known scientists in India in the field of Vaccum Electronic Devices for over three and half decades. His professional achievements span a wide area of vaccum microwave devices ranging from crossed-field and linear-beam devices to present-day gyrotrons. He was awarded a degree of M.Sc (Physics) and Sc Tech (Electronics) from BITS Pilani. He started his professional career in 1966 at Central Electronics Engineering Research Institute (CEERI), Pilani. During this period, indigenous know how was developed for several types of fixed frequency and tunable magnetrons of conventional and coaxial type. He also has several publications and a patent to his credit.

B. DEEPAK KUMAR CHAUDHARY

Pursuing M.TECH, Communication & Signal Processing System

Deepak Kumar Chaudhary was awarded a Bachelor of Electronics & Communication Engineering degree from the Pokhara University, Nepal, in 2008. During this period, his contribution was in Simulation of Digital Audio Broadcasting (DAB). Digital Audio Broadcasting (DAB) is a digital radio technology for broadcasting radio stations. He is currently working towards the M.TECH degree in the Department of Electronics & Communication Engineering at the Jaipur National University, Jaipur, India. His current research interests include: “Application of Multiobjective particle swarm optimization to maximize coverage and lifetime using wireless sensor network”.