Volume 3, Issue 10, October 2013 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

For Achieving Best Connectivity in WSN Using GA and Comparative Analysis

Mr. Rohit Prabhakar, Ms. Palvee, Ms. Manpreet Kaur Gill
Department of ECE,
IITT College of Engineering
Pojewal Distt. Nawanshahr,Punjab. India

Abstract: The popularity of Wireless Sensor Networks (WSN) has increased tremendously in recent time due to growth in Micro-Electro-Mechanical Systems (MEMS) technology. Wireless sensor network (WSN) is a collection of tiny, large number of densely deployed sensor nodes. These sensor nodes are smart and effective which is very powerful and versatile networking where traditional wired and wireless networking is unable to deploy. In WSNs it is very difficult to collect the information in an energy efficient manner. The Efficient-Energy Coverage (EEC) problem is a very important issue in this kind of networks that is used to be considered in order to maximize the network life time. As the size of the network increases, the routing protocol becomes more complex due to the amount of sensor nodes in the network. The sensor nodes in Wireless Sensor Networks are very constrained in memory capability, processing power and battery. In this paper, Genetic Algorithm based routing algorithms have been proposed to solve the Efficient-Energy Coverage (EEC) problem trying to deal with these constrains. The GA generates a whole new path of routing by taking energy as fitness value to judge different path and choose best optimized path whose energy consumption is less as compared to other routing paths. We concluded the result which are obtained by performing experiment on our proposed algorithm GA and comparing its result with Ant Colony Optimization which shows better result as compared to Ant Colony Optimization and the experiments performed are done using Mat lab software.

Keywords: Genetic Algorithm, Ant Colony Optimization, Routing, Energy Optimization, Wireless Sensor Network.

I. Introduction

Wireless sensor networks (WSNs) have gained worldwide attention in last few years, particularly with the availability of Micro-Electro-Mechanical Systems (MEMS) technology which has motivated the development of smart sensors. These sensors are small in size, with limited processing and computing resources. They are cheap in cost as compared to traditional sensors. These sensor nodes can sense, measure, and gather information from the environment and, based on some local decision process, they can transmit the sensed data to the user terminal. Smart sensor nodes are low power devices equipped with one or more sensors, a processor, memory, a power supply, a radio, and an actuator if required. A variety of mechanical, magnetic, thermal, biological, chemical and optical sensors may be attached to the sensor node to measure properties of the environment depending on the application. Since the sensor nodes have limited memory and are typically deployed in difficult-to-access locations, a radio is implemented for wireless communication to transfer the data to a base station (e.g., a laptop) [1].

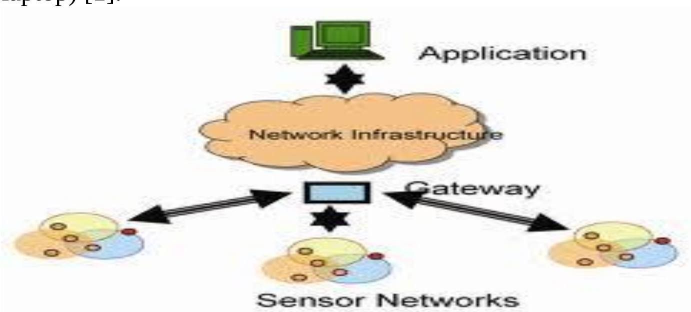

Figure 1: Wireless Sensor Network

II. Literature Survey

In this paper we have discussed various methods for efficient energy coverage in WSN. Energy efficient routing protocol is used to deliver the data in wireless sensor network in where the approach of path break to repair has been discussed. Routing protocol based on Ant colony optimization, where ability of Ants to select the shortest path among few possible paths connecting their nest to a food site is applied for routing. Genetic algorithm based approach for energy efficient routing, where strongest individual survives in a generation. This is used to model after the natural process of evolution as it occurs through offspring. The Ant colony is used to find the shortest path for the routing in WSN.

III. Genetic Algorithm

In the computer science field of artificial intelligence, a genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic (also known as a met heuristic) is routinely used to generate useful solutions to optimization and search problems. Genetic algorithm are used to belong the larger class of evolutionary algorithms(EA), which generate a solution to optimize the problem using techniques inspired by natural evolution such as an inheritance, mutation, selection and crossover.

- **Initialization**

Initially many individual solutions are (usually) randomly generated to form an initial population. The populations sizes are used to depend upon the nature of the problem but it contain thousand of possible solutions. The population was generated randomly and allows the entire range of possible solutions. Occasionally, the solutions may be "seeded" in areas where optimal solutions are likely to be found.

- **Selection**

In each successive generation, a proportion of the existing population was selected to breed a new generation. Individual solution is used to select through a fitness-based process where fitter solutions are used to be selected. In certain selection method rate, the fitness of each solution and preferentially select the best solutions. In other method rate, only a random sample of the population may be time-consuming.

The fitness function is defined the genetic representation and measure the quality of the represented solution. The fitness function is always a problem dependent. In the knapsack problem, once want to maximize the total value of the object that can be used in a knapsack of some fixed capacity. The representation of a solution might be an array of bits where each bit represent a different object and the value of the bit represent whether or not the object is in the knapsack. The every representation is not valid as the size of object may exceed the capacity of the knapsack. The fitness is the sum of all the value of object in the knapsack if the representation is valid or not.

In some problems, it is hard to define the fitness expression. In this case, a simulation may be used to determine the fitness function value of a phenotype or even interactive genetic algorithms are used.

- **Genetic operators**

The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover (also called recombination) and/or mutation. The new solution are used to be produced, a pair of "parent" solutions is selected for breeding from the pool selected previously. By producing a "child" solution using the above method of crossover and mutation, The new solution was created in which typically shares many of the characteristics of its "parents". New parents are selected for each new child and the process continues until a new population of solutions of appropriate size is generated. The reproduction method are based on the use of two parents which are more "biology inspired", some research suggest that more than two "parents" generate higher quality chromosomes.

These processes ultimately result in the next generation population of chromosomes that is different from the initial generation. The average fitness will be increased by the procedure for the population, the best organisms from the first generation are selected for breeding which has a small proportion of less fit solutions for reasons already mentioned above.

Opinion is divided over the importance of crossover versus mutation. There are many references in Fogel that support the importance of mutation-based search. The crossover and mutation are known as the main genetic operators. It is possible to use other operator such as regrouping, extinction or migration in genetic algorithms. The parameter such as the mutation, crossover and population size are used to find reasonable settings for the problem class being worked on.The small mutation rate may lead to genetic drift (which is non-ergodic in nature). The recombination rate is too high to premature convergence of the genetic algorithm. A mutation rate is too high to loss of good solutions unless there is elitist selection. They are theoretical but not yet practical upper and lower bounds for these parameters that can help guide selection.

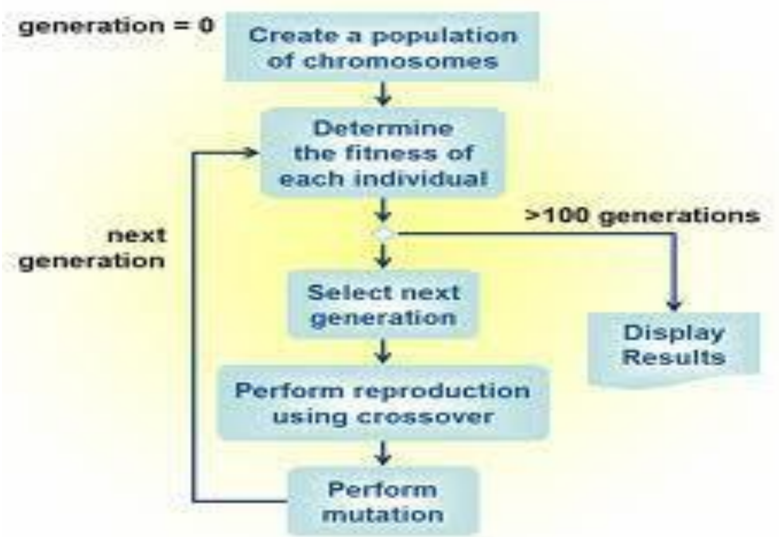

Figure 4: flow diagram of Genetic Algorithm

IV. ANT Colony Optimization

Ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. In recent years, bacterial foraging behaviour has provided rich source of solution in many engineering applications and computational model. It has been applied for solving practical engineering problems like optimal control, harmonic estimation, channel equalization etc. In this thesis, ACO has been used for cluster head selection to provide improved energy efficiency in routing. This section discusses process of cluster head selection using ACO algorithm. The process of cluster head selection involves application of a clustering algorithm. Ant colony optimization (ACO) is a population-based metaheuristic that can be used to find approximate solutions to difficult optimization problems.

In ACO, a set of software agents called artificial ants search for good solutions to a given optimization problem. When we apply ACO, the optimization problem is transformed into the problem of finding the best path on a weighted graph. The artificial ants (here after ants) incrementally build solutions by moving on the graph. The construction process is stochastic and is biased by a pheromone model which has a set of parameters associated with graph components (either nodes or edges) whose values are modified at runtime by the ants.

A) ACO Algorithm

The ant colony optimization algorithm (ACO) is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs. It is a member of ant colony algorithms family. In swarm intelligence methods, it constitutes some meta-heuristic optimizations. Ant Colony Optimization (ACO) is a population-based general search technique for the solution of difficult combinatorial problems which is inspired by the pheromone trail laying behaviour of real ant colonies. The behaviour of ant is exploited in artificial ant colonies for the search of approximate solutions to discrete optimization problems to continuous optimization problems and to important problem in telecommunications such as routing and load balancing. The first algorithm was aiming to search an optimal path in a graph proposed by Marco Dorigo in 1992 in his PhD thesis. It is based on the behaviour of ant seeking path between the colony and a source of food.

The ant colony optimization (ACO) meta-heuristic is a colony of artificial ant are used to cooperate to finding a good solution for difficult discrete optimization problems. The cooperation is a key design component of ACO algorithms. The choice is used to allocate the computational resources to set a relatively simple agents that communicate indirectly by stigmergy. The good solution is an emergent property of the agents' cooperative interaction.

The original idea has been diversified to solve a wider class of numerical problems and several problems have emerged and drawing on various aspects of the behaviour of ants. The main underlying idea are used to loosely inspired by the behaviour of real ants is that of a parallel search over several constructive computational threads based on local problem data and on a dynamic memory structure containing information on the quality of previously obtained result. The collective behaviours are used to emerging from the interaction of the different search threads has proved effective in solving combinatorial optimization (CO) problems.

B) AS (Ant system algorithm):

Ant System (AS) was the first (1991) ACO algorithm. It is an important resides mainly in being the prototype of a number of ant algorithms which have found many interesting and successful applications. Three AS algorithm have been defined in which differ by the way pheromone trails are updated. These algorithm are known as an ant density, ant quantity and ant cycle. In an ant density and ant quantity ant deposit pheromone while building a solution. In ant-cycle ants deposit pheromone after they have built a complete tour.

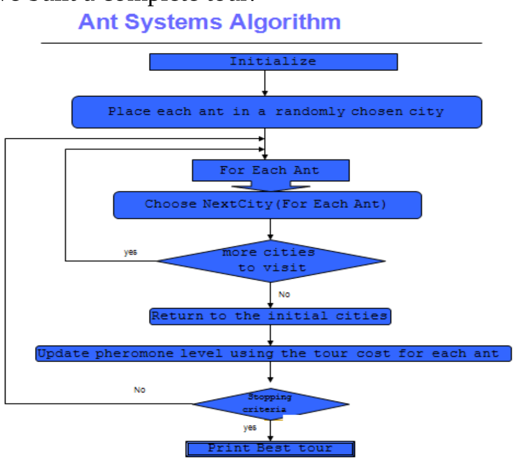

Figure 2:- Ant Systems Algorithm

C) ACS (Ant Colony System):

ACS was the first algorithm inspired by real ant's behaviour. The advantage is used to introduce the ACO algorithm to show the potential of using artificial pheromone and an artificial ant to drive the search of always better solutions for complex optimization problems.

- **Pheromone:**

In ACS, once all ants have computed their tour. AS update the pheromone trail using all the solutions produced by the ant colony. Each edge are used to belong the one of the computed solutions is modified by an amount of pheromone [1] proportional to its solution value. The pheromone of the entire system is evaporate at the end of this phase and the process of construction and update is iterated. In ACS, the best solution was computed since the beginning of the computation is used to globally update the pheromone. In AS, global updating was intended to increase the attractiveness of promising route but ACS mechanism is more effective since it avoids long convergence time by directly concentrate the search in a neighbourhoods of the best tour found up to the current iteration of the algorithm.

D) ACO algorithm for TS problem

Ant Colony Optimization algorithms have been used to produce near-optimal solutions to the travelling salesman problem [6]. The first ACO algorithm was called the Ant system and it was aimed to solve the travelling salesman problem in which the goal is used to find the shortest round-trip to link a series of cities. This algorithm is relatively simple and it is based on a set of ants, each ant making a one of the possible round-trips along the cities. In the traveling salesman problem, a set of cities are given and the distance between each ant is known. The goal is used to find the shortest tour that allows each city to be visited once and only once. At each stage, the ant chooses to move from one city to another according to some rules:

- It must visit each city exactly once;
- A distant city has been less chance of being chosen (the visibility);
- The more intense the pheromone trail laid out on an edge between two cities, the greater the probability that edge will be chosen;
- Having completed its journey, if the journey is short then ant deposits more pheromones on all edges it traversed.

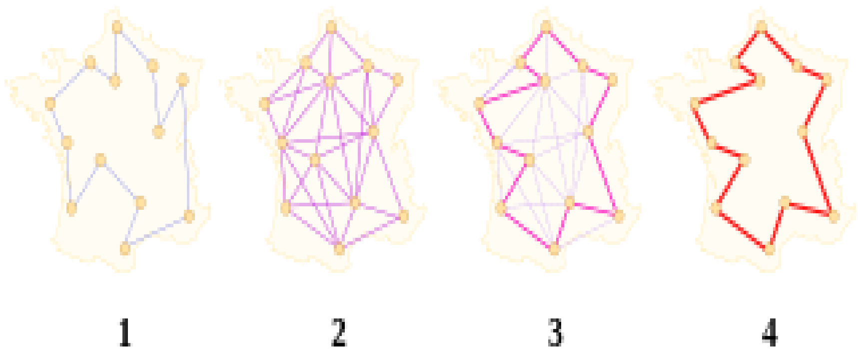

Figure 3:- Travelling Salesman example

The general ant colony algorithm is used to optimize the problem which is described as follow rules:

- Initialization
- Define Pheromone of every move
- Pheromone updating rule
- Ants moving rule
- Stopping rule

1) Initialization

In TSP, suppose N is a set of cities and E is a set connecting edge of two cities. Define a set of paths is w, feasible solutions of TSP, which connecting an initiative city and terminal city through a series of in term city by edges. Ants are placed in cities arbitrarily. In this problem, each ant will choose a path of w according to rules. The numbering of cities from 1 to N if there are N cities. Define (i, j) is an edge of city i and city j.

2) Define Pheromone of every move:

The pheromone's quantity of each ant is based on optimization problem. Every ant is a simple agent to travel through cities. According to probability, an ant chooses one city to move into. This probability is a function of cities' distance and edge's sum pheromone. Every ant has recording cities which ant has been accessed.

3) Pheromone updating rule

Ants leave their pheromone on edges at their every travelling when ants complete one iteration. The sum pheromone of one edge is defined as following:

$$\tau_{i,j}(t+1) = \Delta\tau_{i,j}(t) + (1-\rho)\,\tau_{i,j}(t)$$

$(1 - \rho)$ is persistence rate of previous pheromone . ρ is defined as evaporation of pheromone.

4) Ants moving rule

Ants move from one city to another city according to probability. Cities accessed must be placed in table which defines a set of cities never access of kth ant as allowed k.

5) Stopping rule

There are many conditions for ants to stop their travelling. Such as a number of limited iteration, CPU time limitation or best solution from the above description. we get detail procedure of an ant colony system algorithms like ant-cycle, ant-density and ant-quantity algorithm.

V. Simulation

A) Experimental results of Ant Colony Optimization

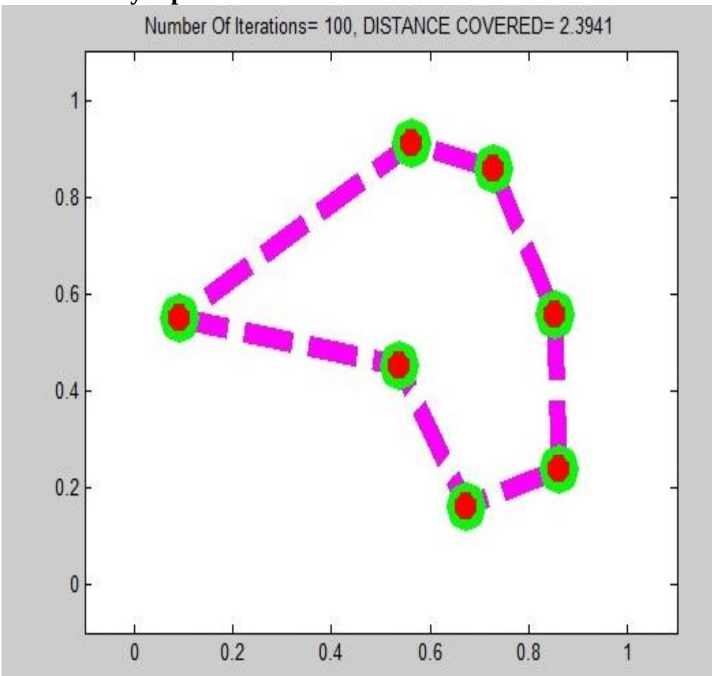

Figure 5:-No. of iteration and distance performed in ACO

B) **Experimental results of Genetic Algorithm**

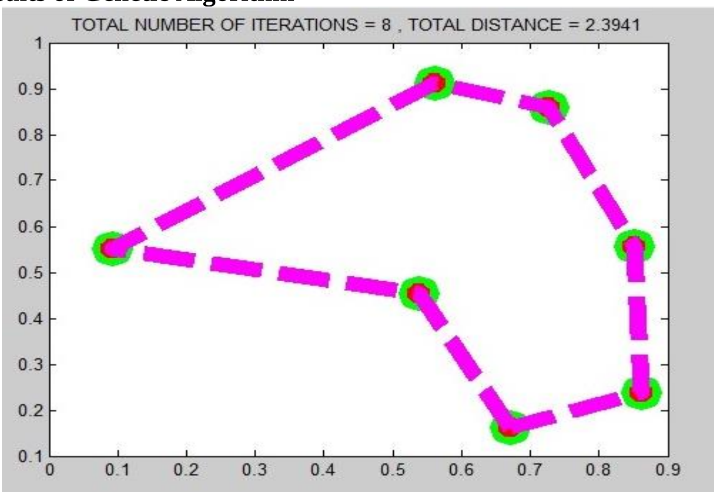

Figure 6:-No. of iteration and distance performed in GA

C) **Comparison between ACO and GA**

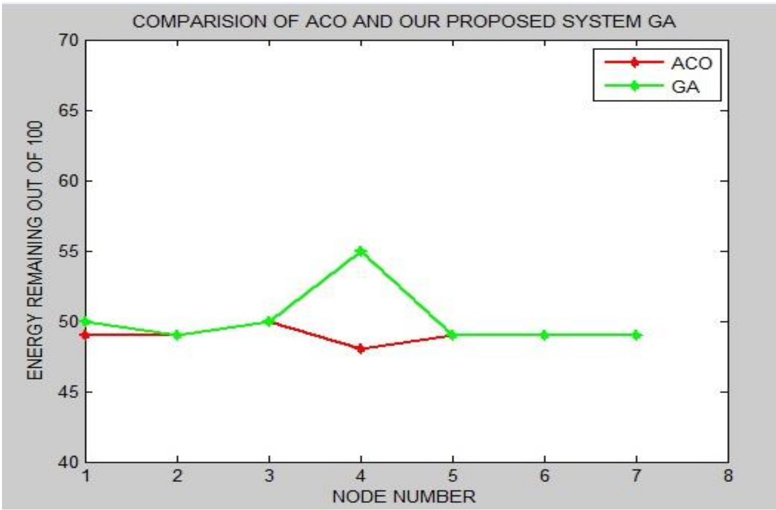

Figure 7:- Comparison between ACO and GA

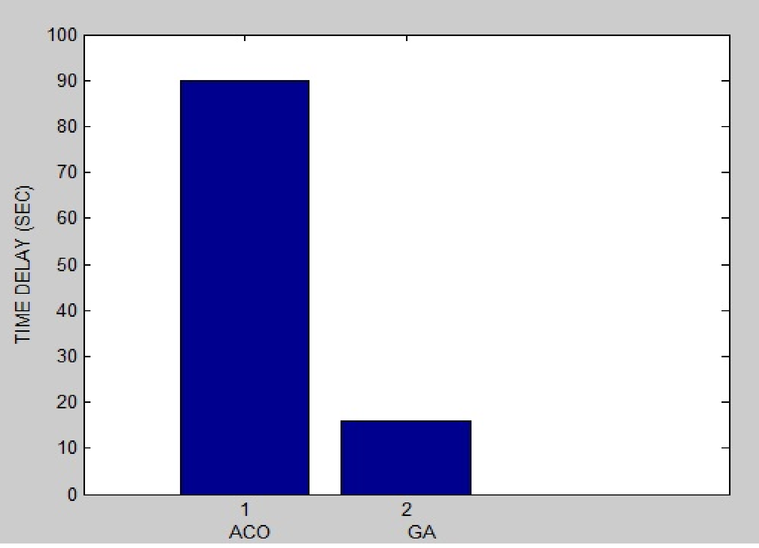

Figure 7:- Time delay between ACO and GA

VIII. Conclusion

In this paper, introduce a novel GA algorithm, which is used to solve the EEC problem. The proposed GA algorithm has new characteristics which are different from conventional ACO algorithms. To achieve the longer lifetime of network and it is important to find maximum number of disjoint subsets of devices in the scheduling method. Then introduce the heterogeneous WSN, which is made by the random selection of the parameters of the probabilistic sensor detection model. We have also proposed a solution to the EEC problem under continuous space and used real numbers as position values of the sensors and the POIs, providing a realistic approach. The Comparison between the Genetic algorithm and ACO are used to improve the energy consumption by sensor node and prolong the lifetime of the network.

Genetic Algorithm (G.A) based cluster head selection has been presented in this paper. It is seen that GA provides better performance than other popular techniques. GA is used to reduce the energy consumption by nodes and prolong the network lifetime of the network. However, the computational complexity of GA for applicability to WSNs still remains a challenge.

References

[1] I. F. Akyildiz, T. Melodia, and K. R. Chowdury, “Wireless multimedia sensor networks: A survey,” IEEE Trans. Wireless Commun, vol. 14, no. 6, pp. 32–39, Dec. 2007.

[2] J.-Y. Heo, J.-M.Hong, and Y.-K. Cho, “EARQ: Energy aware routing for real-time and reliable communication in wireless industrial sensor networks,” IEEE Trans. Ind. Informat., vol. 5, no. 1, pp. 3–11, Feb. 2009.

[3] Q. Li, L. Cui, B. Zhang, and Z. Fan, “A low energy intelligent clustering protocol for wireless sensor networks,” in Proc. IEEE Int. Conf. Ind. Tech., Mar. 2010, pp. 1676–1682.

[4] A. Luntovskyy, V. Vasyutynskyy, and K. Kabitzsch, “Propagation modeling and placement algorithms for wireless sensor networks,” in Proc. IEEE Int. Symp. Ind. Electron., Jul. 2010, pp. 3493–3497.

[5] Y. Z. Zhao, T. N. Nguyen, M. Ma, and C. Miao, “An energy-efficient MAC protocol with adaptive scheduling for wireless sensor networks,”in Proc. IEEE Int. Conf. Ind. Informat., Jun. 2009, pp. 446–461.

[6] S. Nouh, R. A. Abbass, D. A. E. Seoud, N. A. Ali, R. M. Daoud, H. H. Amer, and H. M. ElSayed, “Effect of node distributions on lifetime of wireless sensor networks,” in Proc. IEEE Int. Symp. Ind. Electron., Jul. 2010, pp. 434–439.

[7] Wang,X., Ma,J., & Wang, S. (2009). Parallel energy-efficient coverage optimization with maximum entropy clustering in wireless sensor networks. Journal of Parallel and *Distributed Computing, 69,* 838-847.

[8] Ferentinos,K.P., & Tsiligiridis,T.A. (2007). Adaptive design optimization of wireless sensor networks using genetic algorithms. *Computer Networks, 51*, 1031-1051.