International Journal of Intelligent Information Technologies

April-June 2011, Vol. 7, No. 2

Table of Contents

Research Articles

Wireless Sensor Node Placement Using Hybrid Genetic Programming and Genetic Algorithms

Arpit Tripathi, Indian Institute of Information Technology and Management Gwalior, India

Pulkit Gupta, Indian Institute of Information Technology and Management Gwalior, India

Aditya Trivedi, Indian Institute of Information Technology and Management Gwalior, India

Rahul Kala, Indian Institute of Information Technology and Management Gwalior, India

ABSTRACT

The ease of use and re-configuration in a wireless network has played a key role in their widespread growth. The node deployment problem deals with an optimal placement strategy of the wireless nodes. This paper models a wireless sensor network, consisting of a number of nodes, and a unique sink to which all the information is transmitted using the shortest connecting path. Traditionally the systems have used Genetic Algorithms for optimal placement of the nodes that usually fail to give results in problems employing large numbers of nodes or higher areas to be covered. This paper proposes a hybrid Genetic Programming (GP) and Genetic Algorithm (GA) for solving the problem. While the GP optimizes the deployment structure, the GA is used for actual node placement as per the GP optimized structure. The GA serves as a slave and GP serves as master in this hierarchical implementation. The algorithm optimizes total coverage area, energy utilization, lifetime of the network, and the number of nodes deployed. Experimental results show that the algorithm could place the sensor nodes in a variety of scenarios. The placement was found to be better than random placement strategy as well as the Genetic Algorithm placement strategy.

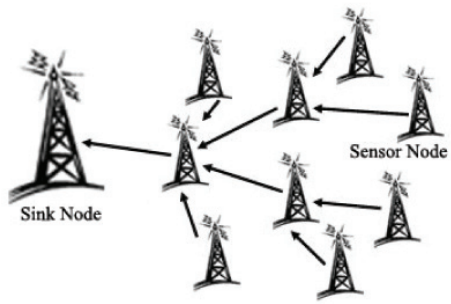
Keyword: *Genetic Algorithms, Genetic Programming, Hierarchical Genetic Algorithms, Node Deployment, Wireless Sensor Networks Optimization*

1. INTRODUCTION

From the last few years, Wireless Sensor Networks (WSN) have come out as the most important networking paradigm, both in terms of their viable prospective and also from a scientific point of view. WSNs are important because of their numerous applications; vary from military use to environmental monitoring (measuring temperature, humidity, & solar radiation etc.) and from wildlife to disaster

DOI: 10.4018/jiit.2011040104

Figure 1. Passage of information from sensor nodes to sink



management. Efficient sensor node deployment is essential to sense, gather and process the data. Therefore, in the last few years there is a shift in research activities in this field. Traditionally sensors nodes were distributed randomly or uniformly due to ease and effortlessness. But random or uniform allotment of sensor nodes is sub-optimal and further leads to the Sink Routing-Hole Problem (SRHP).

Sensor nodes transmit the information to the sink via multiple hops as shown in Figure 1. A node near the sink gets data transmitted by farther nodes as well, in addition to its own data, which is finally transmitted to the sink. Thus a node closer to the sink transmits more data than the ones farther from it. The problem with these networks is maximized if some node near the sink fails (Yunhuai, Ngan, & Ni, 2007). The restriction with WSN is their limited source of energy (as they are composed by battery supplied nodes), concern of the coverage constraint, and the trustworthiness and life-time of network. Whenever WSN designers plan WSN they have to consider these considerations along with number of sensor nodes to wrap the region since wireless sensors are still expensive.

The problem of sensor deployment for an optimal sensor network layout can be solved using either of the two strategies: (i) find the minimum number of sensor nodes of given energy to cover the region with appropriate reliability and a good life time, and (ii) for a given number of sensor nodes, find out a set of points and the power levels which give the best trade-off between coverage area, lifetime and energy utilization of the network with appropriate reliability.

While the first strategy limits the maximum area to be covered, keeping the number of sensor nodes variable; the second strategy keeps the number of sensor nodes fixed, making the other parameters variable. In this paper we propose an algorithm for optimizing the number of nodes, as well as the area, lifetime and energy, by using a hybrid of Genetic Programming (GP) and Genetic Algorithms (GA).

Evolutionary Algorithms are an inspiration from the natural species that develop along with generations. The generation of higher generation by the lower generation is done by the application of Evolutionary operators (Melanie, 1999). The Evolutionary Algorithms may be fundamentally categorized into three heads. These are Genetic Algorithms, Genetic Programming and Evolutionary Strategies (Schwefel, 1975; & Rechenberg, 1973). In Genetic Programming (GP) the individual is usually given a tree-based representation and represents a program, solving the fitness of the individuals may be determined (Fogel, 1992). The Genetic Algorithm (GA) makes use of a bit string or a double vector representation where the various properties of the individual are encoded in a contiguous manner to make a vector or string.

Both GP and GA are widely used for a variety of problems. These algorithms however fail to give good results if the fitness landscapes are very complex or highly dimensional in

nature. The inability of a single evolutionary technique to solve the optimization problem, gives rise to hybridization of two or more evolutionary techniques (Hou, Chen, & Jeng, 2010) for better convergence and optimization (Baffo & Confessore, 2010). In this hybrid algorithm, evolutionary algorithms assist each other to achieve a high degree of optimization in a finite amount of time. Here two evolutionary techniques work in master slave mode (Edwin, Thierens, & Watson, 2004).

The entire problem of node deployment may be fundamentally broken down into parts. The first part of the problem deals with finding the correct connectivity or layout of the WSN. This part of the problem deals with how the various nodes need to be connected to each other and to the sink. It further stresses upon the mechanism by which the various nodes transmit their information to one another, so as to finally reach the sink. The other part of the problem deals with the optimal physical placement of the sensor nodes. This part of the problem computes the exact location in the map where the sensor nodes need to be placed. This part of the problem assumes the connection is already decided. The connection puts a constraint on the range within which a connected node may be placed. The two problems are hence dependent on each other and may not be individually solved.

Increased usage of technology in multiple domains presents a promising future where decision support systems would take over much of tasks being currently done by humans. Automated systems are more efficient in terms of time consuming, work in real time, and give optimal results. Examples include text mining (Qi, Song, Yoon, & Versterre, 2011), power systems (Carrasco, Ternero, Sivianes, Oviedo, & Escudero, 2010). In this paper, we present an intelligent system for wireless sensor node placement which makes wise decision regarding the use of GP to solve the first part of the problem and GA to solve the second part of the problem. GP and GA work in a hierarchical manner with GP as master and GA as the slave. In this paper GA serves as the local search strategy of the GP. The GP fixes some structure of the WSN node deployment. GA is initiated to work over this structure and find the most optimal places in the map for placement of the sensor nodes.

The novelty of the paper is three-fold. Firstly, the region where the sensor nodes may be placed is given in form of a map, which may have any flexible shape or design. It may not strictly be rectangular or circular or any fixed shape as per the problems in literature. Secondly by adjusting the various parameters, we may achieve a clear tradeoff between the optimization of the layout structure (or connectivity) and the region of placement of the nodes. This is done by changing the contributions of the two evolutionary algorithms. Thirdly, to a reasonably high degree, the algorithm is adaptive to the number of nodes and coverage area. The algorithm itself finds out the number of nodes that must be deployed to cover the largest possible area.

This paper is organized as follows. In Section 2, we give a review of some related work. The modeling of the problem is given in Section 3. In Section 4 we discuss the GP framework to solve the problem. Section 5 further discusses the framework of the GA in this hybridized approach of GP and GA. The experimental results are shown in Section 6. The conclusion remarks are given in section 7.

2. RELATED WORK

Yener, Ismail, and Sivrikaya (2007) considered optimization models of WSN subject to distance uncertainty for three classic problems in energy limited WSNs: minimizing the energy consumed, maximizing the data extracted, and maximizing the network lifetime. A characteristic feature of his model was uncertainty which was accounted for using a robust optimization technique. In another work, Shouhong and Ding (2009) considered the issue to approach least square solution for range based positioning. They suggested an algorithm that is based on the equations linearized from range measurement equations and implemented a weighted

least square criterion in a computationally efficient way. Lin and Chen (2008) proposed a distributed approach for supporting the randomized scheduling to provide the full coverage. To proficiently achieve this goal, the authors proposed divided their approach into two main stages, field partition and coverage improvement. The managing problems of these two stages had been transferred to two geometry problems, Voronoi polygon construction and circle covering.

Hou, Chen, and Jeng (2010) studied the problem to optimally deploy new sensors in order to improve the coverage of an existing network. In this approach the best- and worst-case coverage problems that are related to the observability of a path were formulated into computational geometry problems. In this work the authors first proved that there exists a duality between the two coverage problems, and then solved the two problems together. Wang, Xu, Takahara, and Hassanein (2007) formulated a generalized node placement optimization problem that aimed at minimizing the network cost with constraints on lifetime and connectivity. As an optimal solution to this problem is difficult to obtain, a two-phase approach was proposed, in which locally optimal design decisions were taken.

Wang, Yang, Ma, He, and Wang (2008) investigated a design approach for minimizing energy consumption and maximizing network lifetime of a multiple-source and single-sink WSN with energy constraints. Both linear and planar network topologies were considered for network lifetime maximization. For planar single-source and single-sink network topologies they successfully used optimality conditions known as Karush-Kuhn-Tucker to obtain analytical expressions of the best possible network lifetime. For planar network topology a decomposition and combination approach was proposed to compute suboptimal solutions. To deal with large scale planar network, an iterative algorithm was proposed.

Gentile proposed convex linear program which ensures that the estimated link distances between neighboring nodes conform to requisite geometrical constraints. He suggested a distributed algorithm to reconstruct the locations of the sensor nodes from the estimated link distances. Shu, Krunz, and Vrudhula (2006) addressed the problem of minimizing energy consumption in a CDMA-based wireless sensor network (WSN). A comprehensive energy consumption model was proposed. Yener, Ismail, and Sivrikaya (2007) considered a multi-hop sensor network and address the problem of minimizing power consumption in each sensor node locally while ensuring two properties: communication connectivity and sensing coverage. Their work presents Markov model and its solution for steady state distributions.

A lot of work is also done by the use of Evolutionary Algorithms to solve the problem optimal node deployment. Genetic Algorithms find a lot of applications in solving the problem because of their real individual representation and optimizing capabilities (Zhao, Yu, & Chen, 2007; Oh, Tan, Kong, Tan, Ng, & Tai, 2007; Seo, Kim, Ryou, & Kang, 2008). The problem has also been solved with the help of Genetic Programming. Johnson, Teredesai, and Saltarelli (2005 and Khanna, Liu, and Chen (2006) proposed a reduced-complexity genetic algorithm for optimization of multi-hop sensor networks. The goal of the system was to generate optimal number of sensor clusters with clusterheads. Barrett (2007) used Genetic Algorithm for optimizing the sensor node placement in the particular problem of intrusion detection.

The increased amount of research largely results in the growth of the problem complexity that makes it impossible for the simple genetic algorithms to solve. As a result a lot of work is being done for hierarchical implementation of the Genetic Algorithms and other evolutionary techniques, which may many times result in better performance. The model proposed by Kala, Shukla, and Tiwari (2010) is an innovative use of hierarchical GA. Here the authors proposed the use of clustering as a means of dividing the entire fitness landscape into multiple clusters which would be solved by individual slave GAs. The master GA coordinated the location and size of these clusters. The number of clusters

reduced from a large size to a very small size which was referred as the finer to coarser approach that suited the highly complex fitness landscapes.

Gautam and Chaudhuri (2007) also proposed an algorithm for optimization in complex landscapes. This approach divided the search space into a number of sub spaces. These sub spaces were redefined after a unit iteration of the algorithm. The individuals could further migrate between the sub-spaces. Another good approach to problem solving by hybrid evolutionary algorithms is Hierarchical Fair Competition based Genetic Algorithms (HFCGA) (Hu, Goodman, Seo, & Pei, 2002, 2005). In this algorithm the entire population pool is divided into classes as per the fitness value. Every individual competes only with the individuals of the same class, or the same range of fitness values. The individuals may migrate between classes, if they reach the required fitness level.

3. PROBLEM MODELING

Data transmission in WSN takes place through multiple hops. As explained in section 1, inner node transmits more data packets than outer nodes. Assume $T_1, T_2, ..., T_N$ are the life times of nodes where N is the number of nodes. The lifetime of the network is given by equation (1).

$$\text{Life Time of Network} = \text{Min}(T_i) \tag{1}$$

Sensor nodes have distinct output power levels:

$$(OPL_1 < OPL_2 < OPL_3 < ... ... < OPL_i$$

where l is the maximum number of power levels for any node). The transmitting distance $(T_{r1} < T_{r2} < T_{r3} < ... ... < T_{ri})$ of any node depends upon the output power level at which the particular node is operating. A node can choose any one of the available power levels for operation. The transmitting range depends upon the power levels. Hence the summation of the transmitting range of nodes between source and destination should be greater than distance between source and destination.

If a path from source to destination uses k hops then equation (2) must hold true.

Sum of the transmitting Ranges of nodes in the Route ≥ Distance between source and destination:

$$T_{ri} + T_{ri}^1 + T_{ri}^2 + T_{ri}^3 + ... + T_{ri}^k \geq d_{i0} \tag{2}$$

Where T_{ri} is the transmitting range of ith node and $T_{ri}^1, T_{ri}^2, T_{ri}^3 ... T_{ri}^k$ are the transmitting range of the nodes that lie on the route to sink for the node i. This is shown in Figure 2. d_{io} is the distance between the node i and sink.

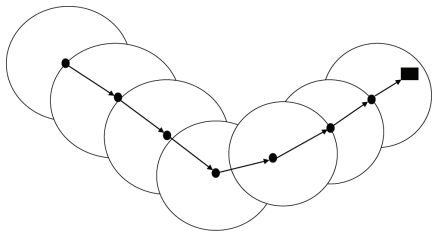
If the distance between two nodes is d then d must be less than or equal to transmission range of node to communicate. So distance between any two nodes can take max value T_{rn}.

Sensor nodes either sense the data or forward it to the next node or sink during the lifetime (T_i). Each node when receives or transmits data uses power from the battery. When battery is discharged, the node is considered as dead. It is possible that data packet chooses a route that has a dead node on its way. Thus death of any node in the network is considered as a death of network.

If E_{ti} is the energy used by i^{th} node for transmitting data packet, E_S *(constant)* is the energy used for sensing and processing, and each node transmit an l-bit message over distance d then energy consumed is given by equation (3) and (4).

$$\begin{aligned} &\text{Transmission Energy } E_t = \\ &(1 * E_{elec}) + (1 * E_{amp} * d^2) \text{ if } d < d_0 \\ &\text{Transmission Energy } E_t = \\ &(1 * E_{elec}) + (1 * E_{amp} * d^4) \text{ if } d > d_0 \end{aligned} \tag{3}$$

Figure 2. Placement of nodes within the transmitting range



$$\text{Receiving Energy } E_S = 1 * E_{elec} \tag{4}$$

Where E_{amp} is constant of the amount of energy consumed during signal amplification in the amplifier and E_{elec} is constant of the amount of energy consumed when data are converted into radio frequency or from radio frequency and d_0 is constant.

The node is usually sleeps and periodically wakes up for transmission of the data packet. If the node wakes up for the time period of T_a for transferring a data packet and transfers total *n* data packets during his life time then number of data packets *n* transmitted is given by equation (5).

$$n = \frac{T_i}{T_a} \tag{5}$$

Here T_i is the lifetime of the node and T_a is the time to sense and transmit a data packet.

If during the lifetime, a node transmits *n* number of messages, then whole amount of consumed energy during the lifetime of node *i* is given by equation (6).

Total Energy Consumed for node i=
Energy consumed to transmit its own pack.
+Energy consumed to transmit pack.
comes from farther nodes

$$E_i = n(E_{t,i} + E_S) + (j-1) * n(E_{t,i} + E_S)$$

$$E_i = jn(E_{t,i} + E_S) \tag{6}$$

Here *j* is the index of node from the farther end of the path belonging i^{th} node and the sink.

If initial energy of the node is E_{int} then node dies when total energy consumed by node is equal to the initial energy of node as given in equation (7).

$$E_{\text{int}} - E_i = 0$$

$$E_{\text{int}} = jn(E_{t,i} + E_S) \tag{7}$$

From equation (5) and (7) we have:

$$E_{\text{int}} = \frac{T_i}{T_a}(E_{t,i} + E_S)$$

$$T_i = E_{\text{int}} \frac{Ta}{j(E_{t,i} + E_S)} \tag{8}$$

If there are *N* numbers of nodes in the network then the total initial Energy is NE_{int}. Given by (9):

$$\text{Energy Efficiency} = \frac{\sum \text{Energy utilized at the time of sink hole}}{\text{Total Energy}} * 100\%$$

$$E_{utl} = \frac{\sum \frac{T_i}{T_a} j(E_{t,i} + E_S)}{N * E\,\text{int}} * 100\% \tag{9}$$

For a relatively small target area where the farthest node is 5 hops away to the sink, less than 8% of the energy is consumed before the system is down. With the size of application area increased, the problem becomes more serious.

For a field with a maximum of 35 hops, when the network fails, only 2% of energy has been spent (Hou, Chen, & Jeng, 2010). Thus random and uniform distribution functions are not suitable for sensor node deployment.

By using path loss models to estimate the received signal level as a function of distance, it become possible to predict the Signal to Noise Ratio (SNR). Log-Normal Shadowing is the most used radio propagation model in WSN application. Measurements have shown that at any value of *d*, the path loss *PL(d)* at particular location is random and distributed log normally(normal in DB) about the mean distance dependent value. This is given by equations (10) and (11):

$$PL_{(d)}[dB\} = PL(d_0) + 10n\log_{10}(\frac{d}{d_0}) + X_\sigma \tag{10}$$

$$\Pr_{(d)}[dBm] = Pt[dBm] - PL_{(d)}[dB] \tag{11}$$

Here *n* is the path loss exponent which indicates the rate at which the path loss increases with distance, d_0 is the close-in reference distance which is determined from measurements close to the transmitter, d is the T-R separation distance, $\Pr_{(d)}$ represents the expected signal strength (in dBm) at the receiver placed at distance *d* (in meters) from the transmitter which delivers $\mathrm{P}t$ as output power (in dBm) and X_σ is zero mean Gaussian distributed random variable (in dB) with standard deviation σ (also in dB). Antenna gains have been considered in channel attenuation $PL_{(d)}$.

For receiving the signal or data packet it is necessary that the SNR is greater than its threshold value $SNR_{\min}$. Probability of successfully receiving the data packets or signals is given by equation (12).

$$P = 1 - \frac{1}{2}e^{\frac{P_t - PL(d_0) + 10n\log_{10}\left(\frac{d}{d_0}\right) - SNR_{\min} - P_n}{\sigma_x\sqrt{2}}} \tag{12}$$

4. GENETIC PROGRAMMING

The first task in the implementation of the algorithm is the layout optimization that is done by the use of GP. The GP module is given a map that is a representation of the areas where the sensor nodes may be deployed. These may be the areas where the wireless sensor is to be provided. The map consists of a grid of size *p x q*. Any cell located at position *(i,j)* in this map M_{ij} is 1 if it is feasible to place a sensor at that location and 0 otherwise.

Layout in the problem of node deployment in WSN means to specify the general connectionist architecture of the system. Here we specify the total number of nodes that are to be placed as well as the connection between the various nodes. A node *i* connected to node *j* means that *i* can transmit data to *j*. The connections must be in such a manner that all nodes must be able to transmit data ultimately to the sink.

4.1. Individual Representation

The first task in the application of GP is the individual representation. Here we use a tree-like representation of the individual. Every leaf or non-leaf node of the tree represents a sensor node. The root of the tree is always fixed to denote the sink. Suppose that a node *i* has children $j_1, j_2, j_3, \ldots j_c$. This means that the sensor nodes $j_1, j_2, j_3, \ldots j_c$ are connected to the sensor node i. In other words there is an inward transfer of information from the sensor nodes $j_1, j_2, j_3, \ldots j_c$ to the sensor node *i*. One such representation is given in Figure 3. Figure 3(a)

shows the phenotype representation and Figure 3(b) shows the genotype representation.

Here we may easily see that the inner sensor nodes are found near the root in the GP individual. The transfer of information from the outer sensor nodes to the inner sensor nodes in phenotype representation of Figure 3(a) is analogous to the transfer of information from leaf nodes to the root node in the genotype representation of Figure 3(b). The entire information must reach the sink that is the central node in phenotype representation and the root node in genotype representation.

We place a few constraints in the individual. The maximum height is fixed to some constant value h_{max}. Also the maximum number of children of any node in the GP individual is fixed to some value c_{max}. Also the maximum total number of sensors or nodes is fixed to some value n_{max}.

The GP individuals, along with storage of the connections also store the physical location where the node is placed. These locations are optimized as well as checked for constraints by the GA. Hence each node of the tree is associated with some x_{rel} and y_{rel} value corresponding to the sensor placement position. The *rel* means that the values are stored relating to the parent node. The root has both x_{rel} and y_{rel} as 0. The absolute value x and y may be calculated from the relative values by iterating from the root in a top to bottom manner. The absolute position of x and y for root are assumed to be known before hand. The placement may never happen at an inaccessible location $(m_{xy} = 0)$. Further the points must lie within the map ($1 \leq x \leq p$, $1 \leq y \leq q$). In case the placement of any node (in the initial individual generation phase or during the operation of any operator) results in the placement violating either of these constraints, it is assumed that further placement of nodes as children of this node would result in placement violating these constraints.

Hence neither this node, nor any of its children are placed. This is because we go radially outward in the phenotype representation, on going down a tree. If a node fails to lie in feasible region, the radially outward nodes would further lie outside the feasible region. This further limits the height of the GP tree. The GP individuals further store the power level at which the node is operating.

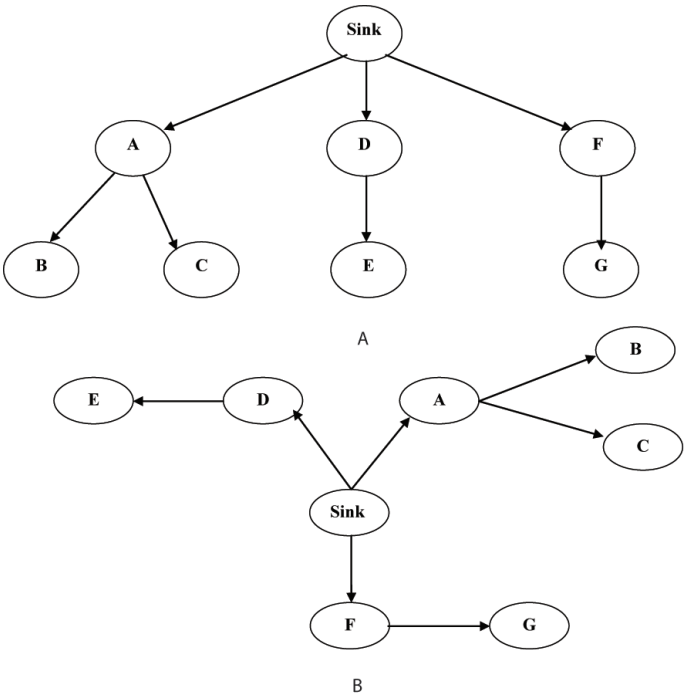
We specify another heuristic rule in the placement of the nodes. An outer node is always placed at the maximum possible distance from the inner node. This maximizes the coverage for the same values of the other objectives. The inner node has some coverage area that depends upon the radius of coverage which again depends upon the power level. The center of the outer node is placed on the rim of this coverage area. Further increasing the distance would lead to loss of connectivity between the two nodes which would be illegal.

4.2. Genetic Operators

The genetic operators are used for the generation of the higher generation of population from the lower generation. In this algorithm rank based scaling with stochastic uniform selection has been used. Crossover is used by mixing two randomly selected sub-trees from the parents and generating two new trees or individuals by their interchange. The interchange must however ensure that the maximum height of any leaf node does not exceed the maximum allowable height. For this we compute the maximum height of both sub-trees and the trees that would be formed upon crossover. In case the crossover may result in crossing of the maximum height threshold, two new sub-trees are selected randomly from the parents.

Mutation in GP is of two types, structural and parametric. The structural mutation tries to change the GP structure. The parametric mutation on the other hand tries to modify the individual property values, stored in each node. In this approach the parametric optimization is carried out by the GA algorithm. The structural mutation tries to select any sub-tree of the selected individual for mutation and replaces it by a completely new and randomly generated

Figure 3(a). Genotype representation of the genetic programming individual; (b). Phenotype representation of the genetic programming individual



sub-tree. The last operator used is elite. This operator transfers the best individual of one generation, directly into the next generation. The parametric mutation has only one role to perform. It randomly selects a node from the entire GP tree. It changes the power level of this tree to some value. In this manner the parametric mutation operator plays a key role in power level optimization. The position of the nodes is not changed in this operation, as this operation is done by the GA.

4.3. Fitness Function

The last major task associated with the GP is to formulate the fitness function. The task of fitness function is to assign values to the individual based on their goodness in solving the problem. This problem is a Multi-Objective Optimization problem where 4 objective functions are to be optimized. The objectives to be optimized are lifetime, energy consumption, coverage area and number of sensor nodes. The fitness function is given by equation (13):

$$Ft = -\alpha_1 * T_{\min} + \alpha_2 * EC - \alpha_3 * A + \alpha_4 * S \quad (13)$$

Here $T_{\min}$ is the lifetime of the node which has minimum lifetime. T_i is the lifetime of the ith node given by equation (8). T_{min} is given by equation (14):

$$T_{\min} = T_j : T_j < T_i \text{ for all } i \neq j \quad (14)$$

EC is the total energy consumed as described by equation (6).

A is the total coverage area of all nodes
S is the total number of sensors.

α_1, α_2 α_3 and α_4 are multi-objective optimization constants.

It may be noted that the coefficients α_1 and α_2 have negative signs to convert the whole problem into minimizing problem.

Before using equation (13) for the fitness evaluation, we need to optimize the physical locations where the nodes may be placed. This is done by the application of GA. The GA optimizes the physical location of the placement of the various nodes and returns the individual with the most optimal fitness. An important characteristic here is that the fitter individual replaced by the GA replaces the original individual whose fitness was being sought in the population pool. Hence next call to the same individual may result in further optimization by the GA.

4.4. Relation Between the Two Evolutionary Algorithms

Here we have used two evolutionary techniques, GP and GA for the optimization of the node deployment. GP is used for the optimization of the layout architecture of the problem. GA is used for the optimization of the physical location where the nodes need to be placed. The GP and GA work in a hierarchical mode in this algorithm. The GP gives the GA in individual which primarily considers the information about the nodal layout architecture. The GA carries out the optimization in the placement of the various nodes represented in the layout architecture.

The optimized individual is returned, whose fitness is used as the GP individual fitness. Further this individual is used to replace the original GP individual in the GP population pool. This is shown in Figure 4.

5. GENETIC ALGORITHM

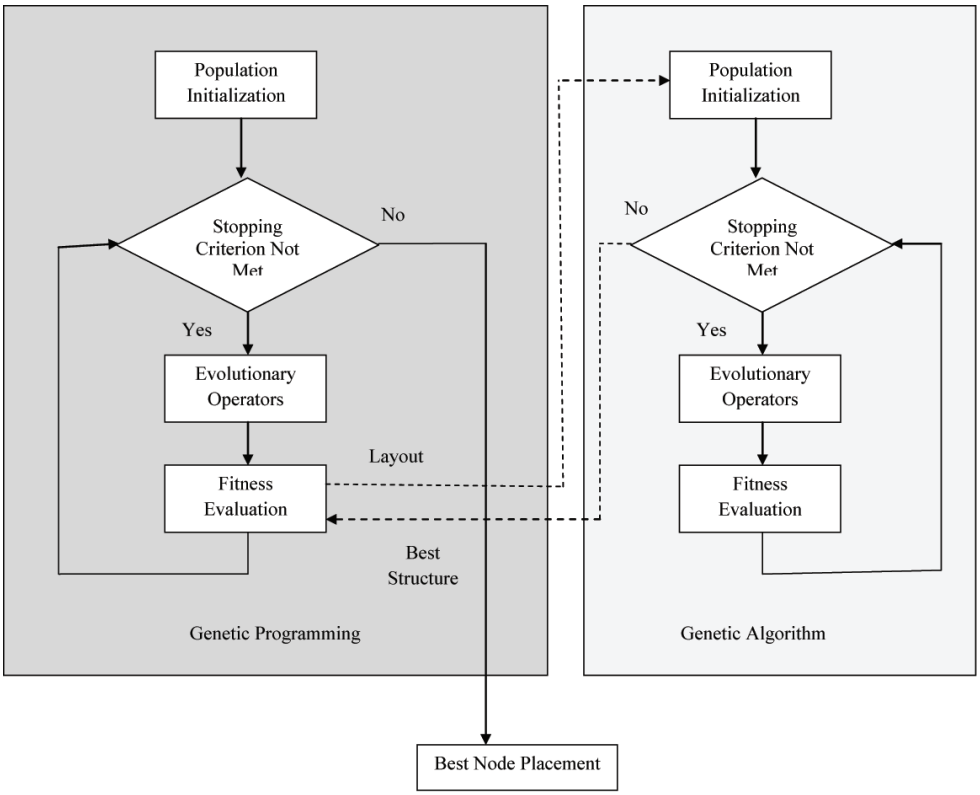
The other part of the solution is the use of GA that fuses with GP for the optimization task. GA is used for optimizing the physical location where the sensor nodes would be located. While implementation of the GP we assume that the structure or the layout of the system has already been done. Each of the nodes in the GP individual has parameters x_{rel} and y_{rel} that stored the relative position of the node relative to its parent. The GA is used for the optimization of these positions. The constraints that the nodes must lay with the map and at accessible regions are valid for GA as well in its placement strategy. Further the child node must be within the coverage area of the parent node. These constraints must be obeyed by every individual of the population at any generation.

The individual representation in case of the GA consists of a string that stores all the relative values. The maximum and minimum values are different for all the locations. This depends upon the power level at which the parent node is operating as fixed by the GP. GA works over a linear architecture, whereas the GP worked over the tree like architecture. We hence need to make a linear representation of the various parameters, based on the supplied GP individual. The final optimized individual again needs to be converted into a tree-like structure, before being returned as the final solution. The conversion from a linear structure to a tree-like structure is also needed before the fitness function calls.

The GA uses the conventional evolutionary operators for the generation of higher generation population from a lower level. These include the stochastic uniform selection, fitness based scaling, weighted crossover, Gaussian mutation and elite. All these in their own mechanism try to influence the evolution of more optimized individuals in the higher generation.

An important operator used here is repair. The various Genetic operators may make the individual disobey some of the constraints. The nodes disobeying the constraints hence need to be repaired by making proper modifications in their values, so that all the constraints are satisfied. This task is performed for all the individuals in the population by the repair operator. If any child lies outside the coverage area of

Figure 4. The general algorithm framework



the parent, it is pulled in by random amount. If however the node lies in the inaccessible areas or outside the map, it is simply deleted. All the children of this node are deleted as well. In this manner the resultant individual completely obeys the stated rules.

An important aspect of the algorithm is the role of the two evolutionary algorithms. At the start we may expect completely randomized structures being generated by the GP which go on becoming optimized as the GP iterations precede. It would not be wise to heavily optimize the initial GP layouts as we know they most probably represent poor layouts. Hence we may weakly optimize these layouts, just to get a rough idea to separate good layouts from poor layouts. However as the algorithm proceeds, the layouts start getting optimized in nature. We are now expected to have better GA optimization to clearly determine the fitness. For these reasons the number of generations of GA is kept variable. Let g_{max} be the maximum number of generations that the GA may have and g_{min} be the minimum number of generations of the GA. The number of generations of GA increases with the iterations of the GP. Let any point of time, the GP be at iteration i. Let us further assume that the GP has maximum iterations of G. The number of generations of GA g is given by equation (15).

$$g = g_{\min} + (g_{\max} - g_{\min})\frac{i}{G} \tag{15}$$

6. RESULTS

The algorithm was implemented on JAVA platform. Two separate modules were made for the GP and the GA. The algorithm parameters like number of individuals, mutation rate, etc. for both these algorithms was mentioned on a separate parameter file. The algorithm took as input a JPEG image which was used to mark out the accusable and the inaccessible areas. The parsing of the image and formulation of the map as an array was done in a separate file. JAVA Applet was used for the display of the final solution in multiple views.

We first tested the algorithm against a number of scenarios. This was done by passing different types of maps as the algorithm input, which had different shapes and styles. The first input that we gave to the algorithm was a circular map. This consisted of a circular region where the nodes were to be placed. The genetic parameters used for the simulation may be divided into two heads of GP parameters and GA parameters. The GP parameters consisted of 120 individuals and 25 generations. At any generation 50% individuals were contributed by crossover operator, 20% by structural mutation operator, 20% by parametric mutation operator, and 10% by elite operator. The mutation rate was kept as 0.06.

The GA consisted of 40 individuals. The number of generations could increase from 5 to 10. Mutation rate was 0.03 and crossover rate was 0.7. Elite count of 2 was used. The maximum nodes were kept as 75 and the maximum children per node were fixed to be 20. The multi-objective optimization parameters were fixed to 0.3, 0.2, 0.8 and 1.0 for energy, lifetime, area and number of nodes respectively.

The simulation using these parameters took about 5 minutes of execution time when simulated on a 3.0 GHz processor with 1 GB RAM. Figure 5(a) shows the coverage for each node. The transfer of information from the various nodes to the sink is shown in Figure 5(b). From figure it may be seen that the top area is not being covered by any node. This is because of the fact that the coverage of this area would mean an increase in the energy as well as nodal cost. These disadvantages would overcome the benefit of increased area. Another observation that can be made is that the nodes are usually of a larger power level. This is again attributed to the fact that the increase in area coverage by larger power levels overcomes the disadvantage of high energy associated with these levels. The multi-objective parameters play a major role in deciding the node placement and their power levels. These are used to lay different stress on different objectives. The value of the different objectives is given in Table 1.

The second experiment was conducted on a square map. The algorithm could place the different sensor nodes anywhere inside the square. The various parameters used were similar to the ones used in the previous experiment, except for the multi-objective weights which had a value of 0.3, 0.2, 0.9 and 0.7 for energy, lifetime, area and number of nodes respectively. The algorithm was allowed to be run for the set number of generations. AT the completion the placement of the nodes was done as per the best fitness individual. Figure 6(a) shows the coverage of the various placed nodes. The transfer of data from all nodes to the sink is given in Figure 6(b).

Here also we observe that the algorithm did not place nodes in the extremes of the square corners. This is because of the penalty of energy and number of nodes. Another observation may be seen that the nodal placement is neither uniform, nor symmetric in nature. This is due to the fact that multiple objectives need to be simultaneously met with and a number of parameters need to be optimized. This would not happen in case the placement is uniform or symmetric. The value of various objectives is given in Table 1.

Based on the same lines, the last experiment was performed on a diamond shaped map. Here also same set of values to various parameters was used. The multi-objective weights were changed to 0.3, 0.2, 1.4 and 0.7 for energy, lifetime, area and number of nodes respectively. Here as well the algorithm was efficiently able to place the various nodes in an optimal manner.

The coverage area and flow of information is shown in Figure 7(a) and 7(b) respectively. The value of the various objective functions is given in Table 1. Here also the algorithm chose not to cover the entire area, but optimally filled a large part of area, so as to maximize area as well as the energy. The placements were non-uniform and non-symmetric in nature.

In all these experiments we observe that the algorithm did not cover some part of the area. This may also be attributed to the characteristic nature of the map which placed a tradeoff between area maximization and the minimization of the energy and number of nodes. In these executions we had assumed that it is not mandatory for these parts of the map to have coverage. However this may not always be the case. Hence we further study the algorithm behavior by adopting an area dominated strategy. This is done by increasing the weight of area and reducing the other weights in the objective function. The coverage in such a case for the circular map is shown in Figure 8 when the weight corresponding to area was increased to 1.8. It may be clearly seen that more nodes are placed that would require more energy. On the contrary a significantly large area is served with very few exceptions.

The evolutionary approach here comprises of the master GP and slave GA. The task of master is to have an overall convergence which decides the final fitness of the individual and hence the most optimal deployment strategy. The fitness of the best individual along with GP generations is shown in Figure 9.

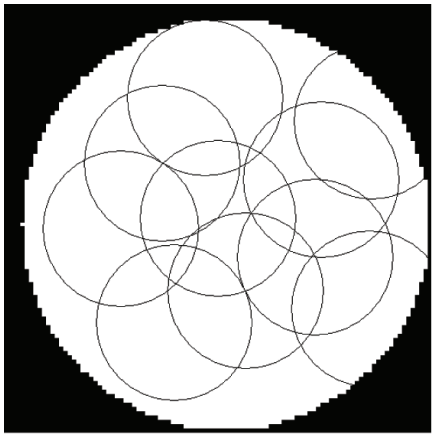
In this algorithm we propped that the number of generations of GA or the slave increase as the algorithm proceeds with the GP iterations. This means a general increase in the time of execution along with GP iterations. We hence study the effect of increasing GA generations on the time.

For this the time needed for the completion of every GP generation is plotted. This is shown in Figure 10. The figure however showcases a characteristic picture where the time of execution first decreases with generations and then constantly increases. Initially the algorithm randomly generates the population that comprise of nodes of multiple power levels.

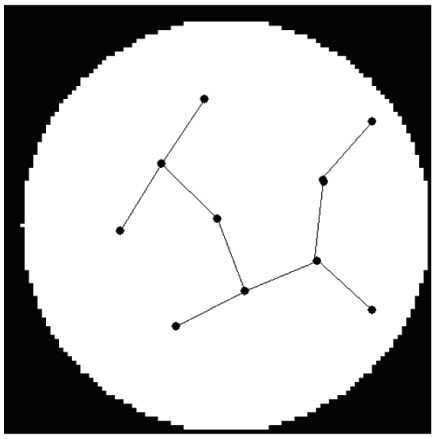
The lower power levels have smaller energy, but cover smaller areas. The initial few generations have a number of nodes corresponding to smaller power levels. As the generations increase, these nodes migrate to higher power levels because of the large increase in area coverage that outperforms the increasing energy. The initial few generations with large number of nodes require more algorithm processing time. However, as the algorithm proceeds, the increasing power levels result in decrease in the total number of nodes. This causes a general reduction in time. Once the nodes have reached significant power level where a tradeoff is met between energy and area coverage, the reduction in number of nodes is not witnessed. Hence the genera increase in time as a result of increasing GA generations is visible in the curve.

The proposed algorithm was able to solve the problem of node deployment effectively and the results were convincing as per the set objective weights. We further analyze the algorithm working in comparison to the other commonly used approaches in literature. The comparison is made with Random Placement, Genetic Programming, and Uniform Node Placement. The performance of the various algorithms is shown in Table 2. In random placement strategy, the nodes were randomly placed onto the map, ensuring the specified constraints are met. A large number of placements were done and the performance of each of these was noted. In each of these placement heuristics were used to maximize area and minimize the lifetime, energy and number of nodes. It may be easily seen that as compared to the proposed algorithm, the random placement technique resulted in a smaller energy and placement of a smaller number of nodes. However the total area coverage in this case was reasonably less as compared to the proposed algorithm. This resulted in the random placement having a reasonably poor value of the objective function as compared to the proposed algorithm. This further emphasizes on the point that the possibility of multiple

Figure 5(a). Area coverage for map 1; (b). Layout and Placement of the nodes for map 1



A



B

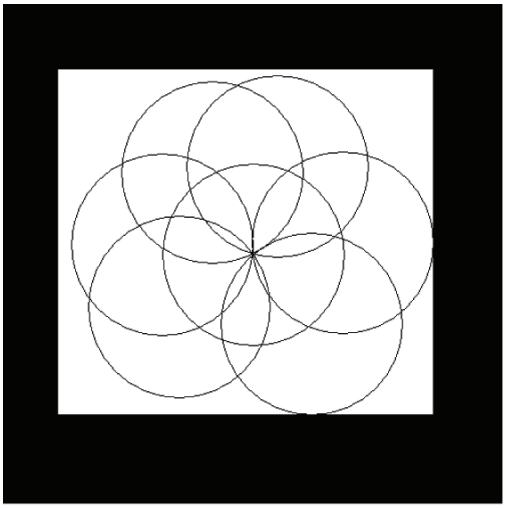
Table 1. The values of objective functions for the various maps

Map	Energy	Lifetime	Area	Number of Nodes	Total
Map 1	6.138×10^{-3}	5.522×10^{-4}	6478.0	10	0.5520
Map 2	3.797×10^{-3}	4.6757×10^{-4}	4504.0	7	0.2157
Map 3	3.6858×10^{-3}	5.5218×10^{-4}	3826.0	6	-0.1139

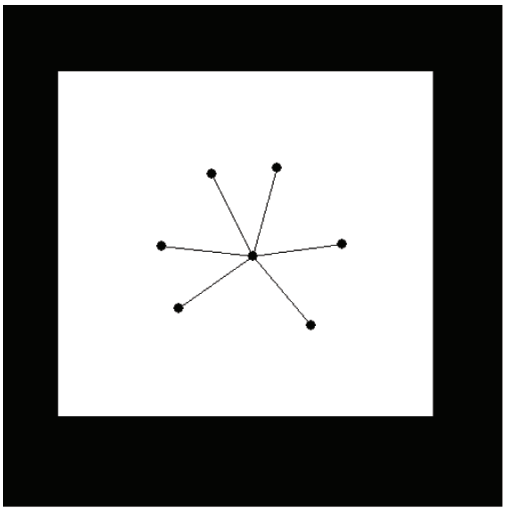
locations of the nodes with different power levels cannot be appreciably done with the use of simple heuristics and requires an intelligent optimization technique.

This algorithm evolved individuals using the same genotype representation as done by the Genetic Programming part of the proposed algorithm. This algorithm was supposed to

Figure 6(a). Area coverage for map 2; (b). Layout and Placement of the nodes for map 2



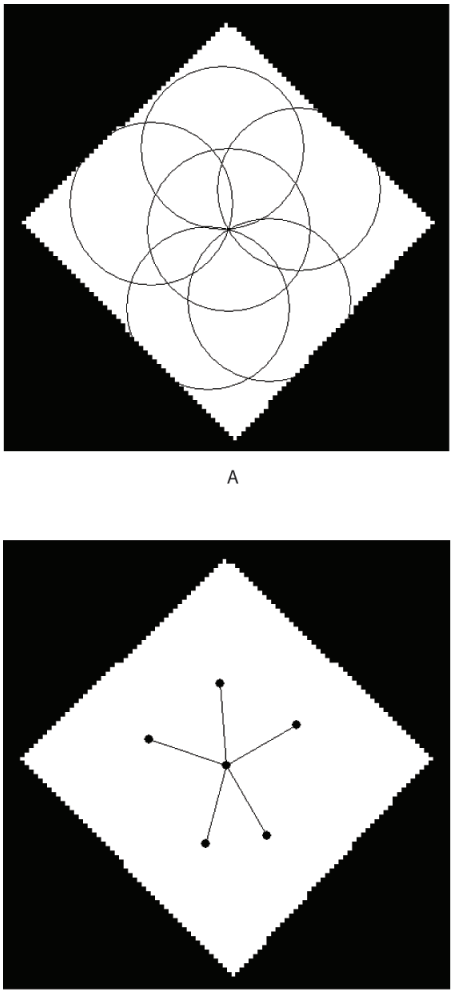
A



B

carry out both structural and parametric optimizations. The algorithm optimized the problem to a fine extent. It resulted in better area coverage as compared to the random algorithm which was however at the cost of placement of a larger number of nodes which resulted in increase of network energy. The value of the net objective function was better as compared to the random node placement. This shows that the algorithm could carry out effective optimizations to increase the coverage area with the least increase in energy and number of nodes. When compared to the proposed algorithm, the genetic programming approach however saw smaller area coverage even with the placement of one more sensor node. The energy was slightly smaller than the proposed algorithm. The proposed algorithm had a worse fitness as compared to the proposed

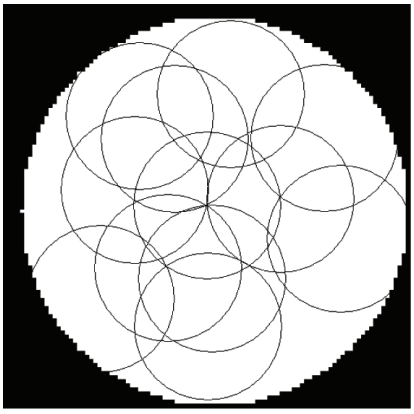
Figure 7(a). Area coverage for map 3; (b). Layout and Placement of the nodes for map 3



algorithm. This shows the inability of a single evolutionary approach to solve the problem that results in hybridization of two evolutionary approaches. Experimental results show that the hybridization results in better performance in optimizing the objective function.

The last algorithm applied was a uniform node placement. Here the various sensor nodes were placed such that the entire area is covered. The uniform placement was done in such a manner that the various sensor nodes are as apart as possible. Each sensor node was operated at the largest power level. This placement technique resulted in a better area coverage that is a natural consequence of the placement strategy. However a very large number of nodes were placed. This resulted in a very large increase in the network energy. The added area coverage did not cope up with the increase in energy. As a result this placement technique had a worse fitness value when compared to the proposed algorithm. This was however better than the random placement and worse than the genetic programming optimized placement.

Figure 8. Area coverage for map 1 with dominance of area objective function



It may be clearly seen that the proposed approach serves out to be the best in optimizing the objective function. The fusion of the GP and GA carried in this approach is able to remove the weaknesses of the individual algorithms to give a better overall result. Hence the approach may be used for a more effective node deployment in different scenarios.

7. CONCLUSION

The ever growing coverage of wireless networks urges the need of better and more intelligent sensor placement strategies. The multi-objective demand along with the need to optimize multiple parameters makes the complete problem difficult to solve by any trivial technique. In this paper we proposed the use of an algorithm hybrid GP and GA for the same.

The GP was used at the master level and carried forward the layout optimization. It tries to evolve the most optimal architecture for the sensor placement. The GP individuals were further optimized by means of the GA that tried to fix optimal positions for the various layouts specified by the GP. This led to an optimization of the complete sensor deployment strategy. The algorithm tried to optimize four objectives namely energy, lifetime, area and number of nodes. The contribution of each of these was specified by the multi-objective weights. The area where the node deployment may take place was kept to be variable and could be specified by the user.

The algorithm was tested against a variety of scenarios by changing the maps. The first experiment was on a circular map, which was followed by experiments on square and diamond shaped maps. In all the experiments we observed that the algorithm was able to have an optimal placement of the various nodes with an optimal layout. In our approach, both the evolutionary algorithms in their own way contributed towards the optimization as per these objectives. Experimental results show that the algorithm placed the various nodes in such a manner that all the objectives are satisfied to some extent. In any of the cases a single factor was not allowed to dominate. Rather the mutual effect of all the objectives was taken. For these reasons the area coverage was never complete. It was made to compromise against the increasing energy and node cost. The different multi-objective weights could be adjusted to get different performances to the algorithm.

A major factor behind the algorithm is to strike a balance between the layout and positional optimization. These are controlled by the two evolutionary techniques of GP and GA respectively. This can be easily done by the adjustment of the evolutionary parameters of both the algorithms. In this approach we therefore made an implementation of deter-

Figure 9. Convergence in genetic programming

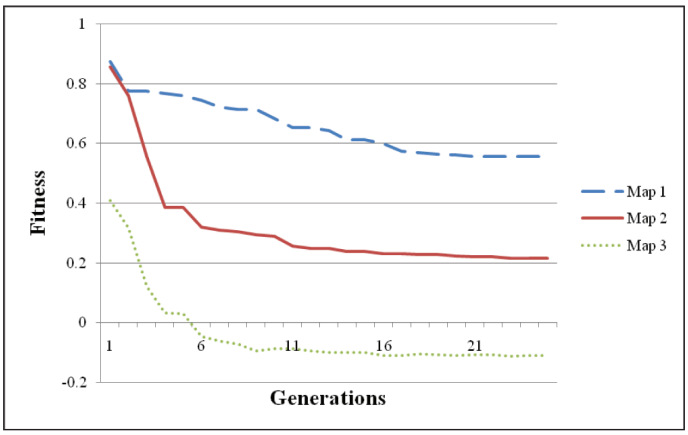


Figure 10. Time of execution for various generations

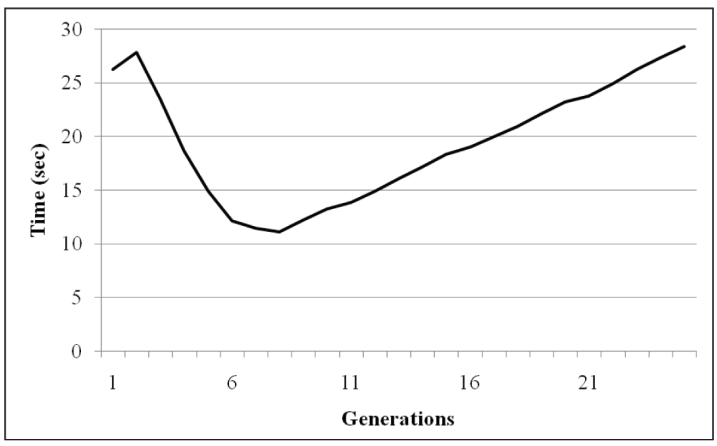


Table 2. Comparative Analysis of proposed algorithm with other algorithms

Algorithm	Energy	Lifetime	Area	Number of Nodes	Total
Hybrid GP and GA	**6.138 x 10^{-3}**	**5.522 x 10^{-4}**	**6478.0**	**10**	**0.5520**
Random	3.5033 x 10^{-3}	5.522 x 10^{-4}	3348.0	6	0.7184
GP	5.2262 x 10^{-3}	5.522 x 10^{-4}	5588.0	11	0.5959
Uniform	9.8208 x 10^{-3}	5.522 x 10^{-4}	7135.0	16	0.6150

ministic adaptation by changing the tradeoff between the two algorithms along with time. The initial few generations are GP dominated, while the domination increases in favor of GA as the algorithm proceeds.

The approach used for solving the problem is an evolutionary approach that is time consuming. More time effective algorithms may be formulated for giving results in shorter time durations. Another problem associated is the scalability. As the dimensionality of the map or the number of placed nodes increase, there is a considerable increase in the search space of the evolutionary algorithm. This lies at both for GP and GA. Very large maps may hence impose a big problem on evolution. The performance may be further improved by framing heuristics for effective guidance of the evolutionary approach. This would result in a considerable reduction of search space at every iteration and an early convergence of the algorithm towards the global minima. All this may be worked over in future.

REFERENCES

Baffo, I., & Confessore, G. (2010). A model to increase the efficiency of a competence-based collaborative network. *International Journal of Intelligent Information Technologies*, *6*(1), 18–30. doi:10.4018/jiit.2010100902

Barrett, S. (2007). Optimizing sensor placement for intruder detection with genetic algorithms In *Proceedings of the IEEE Intelligence and Security Informatics* (pp.185-188).

Carrasco, A., Romero-Ternero, M. C., Sivianes, F., Hernández, M. D., Oviedo, D. I., & Escudero, J. (2010). Facilitating decision making and maintenance for power systems operators through the use of agents and distributed embedded systems. *International Journal of Intelligent Information Technologies*, *6*(4), 1–16. doi:10.4018/jiit.2010100101

Chen, F.-L., & Li, F.-C. (2010). Comparison of the hybrid credit scoring models based on various classifiers. *International Journal of Intelligent Information Technologies*, *6*(3), 56–74. doi:10.4018/jiit.2010070104

Edwin, D. J., Thierens, D., & Watson, R. A. (2004). Hierarchical genetic algorithms. In X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervos, J. A. Bullinaria et al. (Eds.), *Proceedings of the 8th International Conference on Parallel Problem Solving from Nature* (LNCS 3242, pp. 232-241).

Fogel, D. B. (1992). *Evolving artificial intelligence.* Unpublished doctoral dissertation, University of California at San Diego, CA.

Gautam, G., & Chaudhury, B. B. (2007). A distributed hierarchical genetic algorithm for efficient optimization and pattern matching. *Pattern Recognition*, *40*, 212–228. doi:10.1016/j.patcog.2006.04.023

Hou, Y. T., Chen, M. C., & Jeng, B. (2010). An optimal new-node placement to enhance the coverage of wireless sensor networks. *Journal of Wireless Networks*, *16*(4), 1033–1043. doi:10.1007/s11276-009-0186-x

Hu, J., Goodman, E., Seo, K., Fan, Z., & Rosenberg, R. (2005). The hierarchical fair competition (HFC) framework for continuing evolutionary algorithms. *Evolutionary Computation*, *13*(2), 241–277. doi:10.1162/1063656054088530

Hu, J., Goodman, E., Seo, K., & Pei, M. (2002). Adaptive hierarchical fair competition (AHFC) model for parallel evolutionary algorithms. In *Proceedings of Genetic and Evolutionary Computation Conference* (pp. 772-779).

Johnson, D. M., Teredesai, A., & Saltarelli, R. T. (April 2005). Genetic programming in wireless sensor networks. In M. Keijzer, A. Tettamanzi, P. Collet, J. van Hemert, & M. Tomassini (Eds.), *Proceedings of the 8th European Conference on Genetic Programming*, Lausanne, Switzerland (LNCS 3447, pp. 96-107).

Kala, R., Shukla, A., & Tiwari, R. (2010). Clustering based hierarchical genetic algorithm for complex fitness landscapes. *International Journal of Intelligent Systems Technologies and Applications*, *9*(2), 185–205. doi:10.1504/IJISTA.2010.034320

Khanna, R., Liu, H., & Chen, H. (2006). Self organization of sensor networks using genetic algorithms. *International Journal of Sensor Networks*, *1*(3-4), 241–252. doi:10.1504/IJSNET.2006.012040

Lin, J. W., & Chen, Y. T. (2008). Improving the coverage of randomized scheduling in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *7*(12), 4807–4812. doi:10.1109/TWC.2008.070933

Melanie, M. (1999). *An introduction to genetic algorithms*. Cambridge, MA: MIT Press.

Oh, S. C., Tan, H., Kong, F. W., Tan, Y. S., Ng, K. H., Ng, G. W., & Tai, K. (2007). Multi-objective optimization of sensor network deployment by a genetic algorithm. In *Proceedings of the IEEE Congress of Evolutionary Computation* (pp. 3917-3921).

Qi, Y., Song, M., Yoon, S.-C., & Versterre, L. (2011). Combining supervised learning techniques to key-phrase extraction for biomedical full-text. *International Journal of Intelligent Information Technologies*, *7*(1), 33–44. doi:10.4018/jiit.2011010103

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Stuttgart, Germany: Frommann-Holzboog-Verlag.

Schwefel, H.-P. (1975). *Evolutionsstrategie and numerische Optimierung.* Unpublished doctoral dissertation, Technische Universitat Berlin, Germany.

Seo, J. H., Kim, Y. H., Ryou, H. B., & Kang, S. J. (2008). Agenetic algorithm for sensor deployment based on two-dimensional operators. In *Proceedings of the ACM Symposium on Applied Computing* (pp. 1812-1813).

Shouhong, Z., & Ding, Z. (2009). A simple approach of range-based positioning with low computational complexity. *IEEE Transactions on Wireless Communications*, *8*(12), 5832–5836. doi:10.1109/TWC.2009.12.090905

Shu, T., Krunz, M., & Vrudhula, S. (2006). Joint optimization of transmit power-time and bit energy efficiency in CDMA wireless sensor networks. *IEEE Transactions on Wireless Communications*, *5*(11), 3109–3118. doi:10.1109/TWC.2006.04738

Wang, H., Yang, Y., Ma, M., He, J., & Wang, X. (2008). Network lifetime maximization with cross-layer design in wireless sensor networks. *IEEE Transactions on Wireless Communications*, *7*(10), 3759–3768. doi:10.1109/T-WC.2008.070079

Wang, Q., Xu, K., Takahara, G., & Hassanein, H. (2007). Device placement for heterogeneous wireless sensor networks: minimum cost with lifetime constraints. *IEEE Transactions on Wireless Communications*, *6*(7), 2444–2453. doi:10.1109/TWC.2007.05357

Yener, B., Ismail, M. M., & Sivrikaya, F. (2007). Joint problem of power optimal connectivity and coverage in wireless sensor networks. *Journal of Wireless Networks*, *13*(4), 537–550. doi:10.1007/s11276-006-5875-0

Yunhuai, L., Ngan, H., & Ni, L. M. (2007). Power-aware node deployment in wireless sensor network. *International Journal of Distributed Sensor Networks*, *3*(2), 225–241. doi:10.1080/15501320701205597

Zhao, C., Yu, Z., & Chen, P. (2007). Optimal deployment of nodes based on genetic algorithm in heterogeneous sensor networks. In *Proceedings of the International Conference on Wireless Communication Networking and Mobile Computing* (pp. 2743-2746).

Arpit Tripathi is a student of Integrated Post Graduate Course (BTech and MTech in Information Technology) at Indian Institute of Information Technology and Management, Gwalior. His research interests are Wireless Networks, Sensor Networks, Target Tracking and Information Security. He is recipient of the Best in State award for The Great Mind Challenge-2009, a nation–wide software contest, organized by IBM, India. He has been selected for Final round of Nasscom Social Innovation Honours-2011, organized by Nasscom, India.

Pulkit Gupta is a student of Integrated Post Graduate Course (BTech and MTech in Information Technology) at Indian Institute of Information Technology and Management, Gwalior. His research interests are Computer Networks, Wireless Sensor Networks, Reprogramming and Scheduling Algorithms. He has worked as research scholar at National Mobile Communication Research Laboratory (NCRL), South-east University, China. He is recipient of the Best in State award for The Great Mind Challenge-2009, a nation–wide software contest, organized by IBM, India.

Aditya Trivedi is a Professor in the Information and Communication Technology (ICT) Department at Indian Institute of Information Technology and Management, Gwalior, India. He has about 20 years of teaching experience. He obtained his doctorate (PhD) from IIT Roorkee in the area of Wireless Communication Engineering. He received his bachelor degree in Electronics Engg. from the Jiwaji University. He did his M.Tech. (Communication Systems) from Indian Institute of Technology (IIT), Kanpur. His teaching and research interest include Digital communication, CDMA systems, Signal processing, and Networking. He has published around 50 papers in various national and international journals/conferences. He is a fellow of the Institution of Electronics and Telecommunication Engineers (IETE). Dr. Trivedi is a reviewer of IEEE and Springer journals. He is also a reviewer of Tata Mc-Graw Hill books: Fundamentals of Signals and Systems by Michael J Roberts, and Mobile and Cellular Communication. He has been awarded "Bharat Jyoti Award 2011" and was given the IETE's K.S. Krishnan Memorial Award for best system oriented paper in 2007.

Rahul Kala did his Integrated Post Graduate (BTech and MTech in Information Technology) degree at Indian Institute of Information Technology and Management, Gwalior in 2010. He is currently pursuing PhD in Robotics from School of Cybernetics, University of Reading, UK. His areas of research are hybrid soft computing, robotic planning, autonomous vehicles, biometrics, pattern recognition and soft computing. He is the author of two books and has published about 50 papers in various international and national journals/conferences. He also takes a keen interest toward free/open source software. He is recipient of Commonwealth Scholarship 2010 (UK) and winner of Lord of the Code Scholarship Contest organised by KReSIT, IIT Bombay and Red Hat. He secured All India 8th position in Graduates Aptitude Test in Engineeging-2008 Examinations.

CALL FOR ARTICLES

International Journal of Intelligent Information Technologies

An official publication of the Information Resources Management Association

The Editor-in-Chief of the *International Journal of Intelligent Information Technologies* (IJIIT) would like to invite you to consider submitting a manuscript for inclusion in this scholarly journal.

MISSION:

The advent of the World Wide Web has sparked renewed interest in the area of intelligent information technologies. There is a growing interest in developing intelligent technologies that enable users to accomplish complex tasks in web-centric environments with relative ease, utilizing such technologies as intelligent agents, distributed computing in heterogeneous environments, and computer supported collaborative work. The mission of the *International Journal of Intelligent Information Technologies* (IJIIT) is to bring together researchers in related fields such as information systems, distributed AI, intelligent agents, and collaborative work, to explore and discuss various aspects of design and development of intelligent technologies. This journal provides a forum for academics and practitioners to explore research issues related to not only the design, implementation and deployment of intelligent systems and technologies, but also economic issues and organizational impact. Papers related to all aspects of intelligent systems including theoretical work on agent and multi-agent systems as well as case studies offering insights into agent-based problem solving with empirical or simulation based evidence are welcome.

ISSN 1548-3657
eISSN 1548-3665
Published quarterly

COVERAGE/MAJOR TOPICS:

- Agent-based auction, contracting, negotiation, and ecommerce
- Agent-based control and supply chain
- Agent-based simulation and application integration
- Cooperative and collaborative systems
- Distributed intelligent systems and technologies
- Human-agent interaction and experimental evaluation
- Implementation, deployment, diffusion, and organizational impact
- Integrating business intelligence from internal and external sources
- Intelligent agent and multi-agent systems in various domains
- Intelligent decision support systems
- Intelligent information retrieval and business intelligence
- Intelligent information systems development using design science principles
- Intelligent Web mining and knowledge discovery systems
- Manufacturing information systems
- Models, architectures and behavior models for agent-oriented information systems
- Multimedia information processing
- Privacy, security, and trust issues
- Reasoning, learning and adaptive systems
- Semantic Web, Web services, and ontologies

Ideas for Special Theme Issues may be submitted to the Editor-in-Chief.

Please recommend this publication to your librarian. For a convenient easy-to-use library recommendation form, please visit: http://www.igi-global.com/IJIIT and click on the "Library Recommendation Form" link along the right margin.