Energy efficient clustered deployment of Wireless Sensor Network using Particle Swarm Optimization

by

Pranav Khanna

under the guidance of

Mr.Vinod Kumar

A thesis submitted in partial fulfillment of the requirements of the Delhi Technological University for the award of a degree of

Masters of Technology

(Computer Technology and Applications)

June 2012

DELHI TECHNOLOGICAL UNIVERSITY (DTU)

(Formerly Delhi College Of Engineering)

CERTIFICATE

DELHI TECHNOLOGICAL UNIVERSITY
(Govt. of National Capital Territory of Delhi)
BAWANA ROAD, DELHI – 110042

Date: ___________________

This is to certify that the thesis entitled '**Energy efficient clustered deployment of Wireless Sensor Network using Particle Swarm Optimization '**done by **Pranav Khanna (12/CTA/2010),** for the partial fulfillment of the requirements for the award of the degree of Masters of Technology in Computer Technology and Applications, is an authentic work carried out by him under my guidance. The matter embodied in this thesis has not been submitted earlier for the award of any degree or diploma to the best of my knowledge and belief.

Project Guide:

Mr.Vinod Kumar

Associate Professor, Department of Computer Engineering

Delhi Technological University, Delhi 110042

ACKNOWLEGEMENT

I take this opportunity to express my profound sense of gratitude and respect to all those who have helped me throughout the duration of this thesis, with special thanks to my colleague Mr. Sanjay Bisht for supporting me throughout the work of project.

It gives me a great pleasure to express my profound gratitude to my project guide **Mr. Vinod Kumar,** Associate professor**,** Department of Computer Engineering, Delhi Technological University, for his valuable and inspiring guidance throughout the progress of this project. At the same time, I would like to extend my heartfelt thanks to **Dr. Daya Gupta,** Head of the department, Department of Computer Engineering, Delhi Technological University, for keeping the spirits high and clearing the visions to work on the project.

Pranav Khanna

Master of Technology (Computer Technology and Application)

12/CTA/2k10

Delhi Technological University

Abstract

Wireless Sensor Networks (WSN), is a group of low cost, low power, multifunctional and small size distributed networked sensors. These sensors work together to sense the environment with a little or no human intervention. These sensors are randomly deployed in the sensor field which brings the coverage problem. It is a unique problem and in maximizing coverage, the sensors need to be placed in a position such that the sensing capability of the network is fully utilized to ensure high quality of service. This can be achieved with minimum number of sensor nodes having maximum coverage in the network and the nodes are within the communication range.
Particle swarm algorithm can be used to find the optimal positions of the sensors to determine the best coverage. PSO has been successfully used in numerous engineering applications like in training of neural networks to identify Parkinson's disease, extraction of rules from fuzzy networks, image identification, optimization of electric power distribution networks, structural optimization, inhabitant monitoring, environmental monitoring, monitoring deep oceans currents, smart home building and military applications among many others. This algorithm is an optimization technique which belongs to the fertile paradigm of swarm intelligence. It is a derivative free and is a very efficient global search algorithm with few algorithm parameters

In this Project , we have tried to overcome the sensor placement problem in 3- dimensional terrain considering the problem of covering maximum volume with energy efficiency (using minimum amount of energy) to provide larger coverage and longevity to the network.

The research is done in phases where in first relay structures are considered and then cluster structured of network are introduced to further reduce the dissipation of energy . PSO and its variants (APSO and DPSO) are successively applied to search for optimal placement of nodes ,thus providing a comparison/comparative study between the three.

PAPER PUBLICATIONS

In context to the current research the following paper were accepted and published in journals:

- Vinod Kumar, Pranav Khanna and Sanjay Bisht. Article: Adaptive PSO based Algorithm for Optimal WSN Deployment in 3 Dimensional Terrain. *IJCA Proceedings on National Conference on Advancement of Technologies - Information Systems & Computer Networks (ISCON - 2012)* ISCON(2):1-6, May 2012. Published by Foundation of Computer Science, New York, USA.
- Vinod Kumar, Pranav Khanna and Sanjay Bisht. Article: Adaptive PSO based Algorithm for Optimal WSN Energy Efficient Deployment in 3 Dimensional Terrain. *IOAJ Proceedings on International Conference on Computer Science and Engineering (ICCSE-2012)*:19th May 2012. Published by IPM pvt ltd. Interscienec Campus.

Table Of Content

List Of Figures

List Of Tables

Chapter 1: Introduction

1.1 Objective

The objective of this thesis is to design an algorithm using particle swarm optimization techniques for energy efficient clustered deployment of wireless sensor nodes in 3-dimensional terrain , keeping in account the constraints like limited energy , communication range and terrain , thus providing maximum volume coverage with increased longevity of network over the terrain with limited number of sensors.

Currently there have been numerous work on clustered deployment of sensor nodes using various techniques , most of them solves the problem in 2-dimensional terrain , not considering the 3-dimensional constraints of actual world , thus providing Solution in considering only 2-D constraints[2,4]. Others have been related to solving issue like creating random or Self clusters and generating minimum energy communication algorithms for sensor nodes.More Solutions have been given using algorithm like Genetic Algorithm[1,3]. Almost none provides a solution wherein we can specify x,y coordinates of terrain for placement of limited number of sensors to cover as much as maximum volume possible.

In Current thesis , PSO with its variants(APSO & DPSO) have been used to provide the solution for the problem , and thus even provide a comparative study between the three.

1.2 Problem Statement

We want to design an algorithm using particle swarm optimization techniques for energy efficient clustered deployment of wireless sensor nodes in 3-dimensional terrain , to provide best deployment solution for ,with limited number of sensors to cover maximum volume possible. Provide the best particle over iterations as output as the solution. Even Comparative study between the three variants is provided.

The Desired algorithm must be generic , and can be customized for nay number of sensors , sensors specification or terrain. It must also be adaptable with the changing needs and requirements.

1.3 Motivating Factor

The motivating factor of this thesis is two folds , first extending the research into deployment of sensors in 3-dimensional terrain for maximum volume coverage with limited number of sensors . Secondly , using bio-derived technique like PSO and its variants for the search of best particle for deployment and providing a comparison between the three.

Wireless sensor networks are technology of present and future , loads of application have already been built and many are under development , today WSN find its application in various domains ranging from Military , to industry , from sniping to sensing of various deterministic factors .

The basic idea of wireless sensors is to spread the nodes all over terrain , and communicate information together by sensing different factors like temperature , moisture , heat etc of the region and transmitting it to a sink node through intermediate nodes.

Since these sensors nodes are costly , thus at various places keeping the sensors nodes minimum is critical for cost effectiveness , thus our algorithm provides a solution wherein with limited number of sensors maximum volume with consideration of energy constraints can be provided. With the given number of sensors it tries to find the best solution wherein placing the sensors at we can provide the maximum volume covered with minimum energy dissipated.

So in this thesis we have tried to explore a new and efficient way of deployment of sensor.

1.4 Organization of the dissertation

This thesis work is organized as follows:

Chapter 1 deals with providing the objective, problem statement, motivation of undertaking this research work as well as organization of this dissertation.

Chapter 2 gives the introduction about Wireless sensor networks , there application , and issue related to wirless sensor networks.

Chapter 3 introduces Particle Swarm Optimization and its variants.

Chapter 4 Presents the modeling , assumptions and psuedoocode functions for various calculations

Chapter 5 Describes the relay Approach , used earlier in this Research and provides simulation results using various algorithms (PSO,APSO,DPSO)

Chapter 6 Describes the Clustered Approach , used earlier in this Research and provides simulation results using various algorithms (PSO,APSO,DPSO)

Chapeter 7 Comparitive study

Chapter 8 Conclusion And future work

Chapter 2: Wireless Sensor Networks

2.1 WSN

A **wireless sensor network (WSN)** consists of spatially [5] distributed autonomous sensors to *monitor* physical or environmental conditions, such as temperature, sound, vibration, pressure, humidity, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, also enabling *control* of sensor activity. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer applications, such as industrial process monitoring and control, machine health monitoring, and so on.

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from a few to hundreds of dollars, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

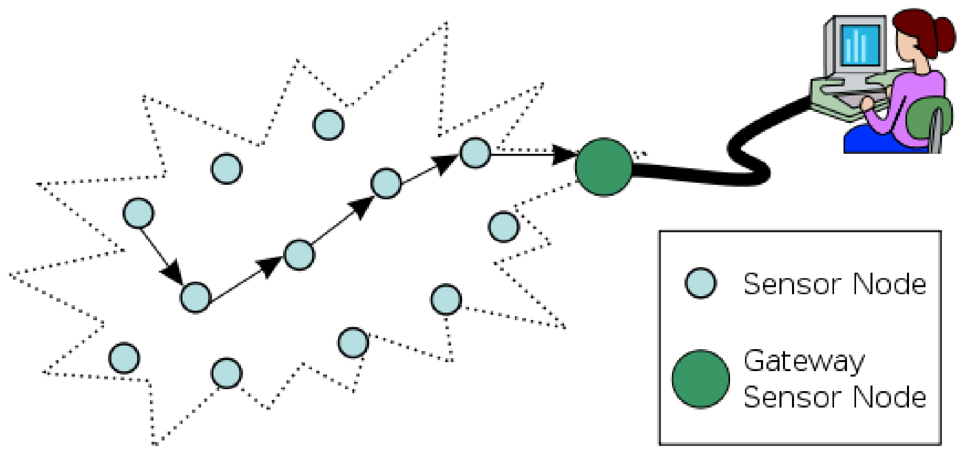

Figure 2.1 Wireless Sensor Network Diagram with Gateway Sensor

2.2 Characteristics

The main characteristics of a WSN include

- Power consumption constrains for nodes using batteries or energy harvesting
- Ability to cope with node failures
- Mobility of nodes
- Communication failures
- Heterogeneity of nodes
- Scalability to large scale of deployment
- Ability to withstand harsh environmental conditions
- Ease of use
- Power consumption

Sensor nodes can be imagined as small computers, extremely basic in terms of their interfaces and their components. They usually consist of a *processing unit* with limited computational power and limited memory, *sensors* or MEMS (including specific conditioning circuitry), a *communication device* (usually radio transceivers or alternatively optical), and a power source usually in the form of a battery. Other possible inclusions are energy harvesting modules, secondary ASICs, and possibly secondary communication devices (e.g. RS-232 or USB).

The base stations are one or more components of the WSN with much more computational, energy and communication resources. They act as a gateway between sensor nodes and the end user as they typically forward data from the WSN on to a server.

2.3 Application of Wireless Sensor Networks

Some of the applications of WSN are:

- Industrial control and monitoring
- Home automation and consumer electronics
- Security and military sensing
- Asset tracking and supply chain management
- Intelligent agriculture and environmental sensing
- Health monitoring

2.4 Issues With Wireless Sensor networks:

Wireless Sensors networks had many issues related to them , thus efficient deployment of network is necessary , some of the issues to be considered are:

- Node deployment : deployment of nodes for maximum volume coverage
- localization ,and energy-aware clustering : since nodes have very less amount of energy stored , hence energy aware clustering is of importance.
- data aggregation are optimization problems in WSN

These Problems pose an optimization challenge while deployment of network.

Properties posing technical challenges are:

- ad-hoc deployment
- dynamic topology ,
- spatial distribution and constraints in bandwidth ,
- memory
- computational resources and energy.

Chapter 3: Particle Swarm Optimization

3.1 Introduction

In computer science, **particle swarm optimization** (**PSO**)[6] is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. Each particle's movement is influenced by its local best known position and is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. This is expected to move the swarm toward the best solutions.

PSO is originally attributed to Kennedy, Eberhart and Shi[8] and was first intended for simulating social behaviour, as a stylized representation of the movement of organisms in a bird flock or fish school. The algorithm was simplified and it was observed to be performing optimization

3.2 Description

PSO is a bio-derived algorithm , like Genetic algorithm and simulated Annealing its derived from the nature and is based on the collective behavior of flock of birds and school of fish in searching food.

Figure 3.1 a) Swarm of Bird searching for food , Figure3.1 b) School of Fish searching for food

“ How can birds or fish exhibit such a coordinated collective behavior?”

Study of these techniques give us an efficient algorithm which explores the given domain by randomized initialization and iterative search to find the best possible solution.

Steps are described as :

The system is initialized with a population of random solutions and searches for optima by updating generations[7]. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles.

Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called *pbest*. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called*lbest*. when a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p*best* and *lbest* locations (local version of PSO). Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations.

Basic Algorithm :

1. Create a ‘population’ of agents (called particles) uniformly distributed over X.
2. Evaluate each particle’s position according to the objective function.
3. If a particle’s current position is better than its previous best position, update it.
4. Determine the best particle (according to the particle’s previous best positions).
5. Update particles’ velocities according to

$$V_{id}^{(t+1)} = w\, V_{id}^{(t)} + c_1 r_1 (p_{id} - x_{id}^{(t)}) + c_2 r_2 (p_{gd} - x_{id}^{(t)})$$

To control the effect of “best” factors of particles

- C1 – cognitive factor (range between 1 and 4)
- C2 – Social factor (range between 1 and 4)

w – is inertia weight which shows the effect of previous velocity vector on the new vector.

rand( ) values range from 0 to 1.

v (t) i is the speed of the ith particle in the t th iteration.

The second part of Eq. thinks about particle own flyingexperience, represents cognition component. The third part considers group flying experience, represents the collaboration among particles giving the social component

6.) Move particles to their new positions according to

$$x_{id}^{t+1} = x_{id}^{t} + V_{id}$$

7. Go to step 2 until stopping criteria are satisfied.

Searching Diagram of PSO

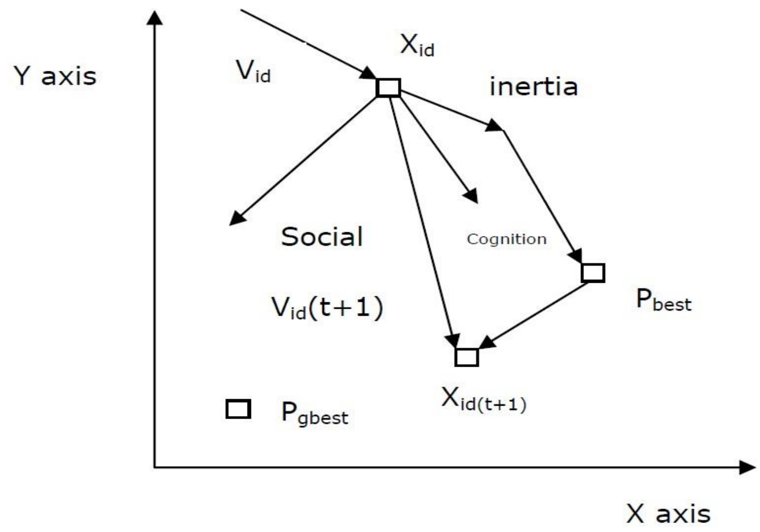

Figure 3.2 Searching Diagram of Particle Swarm Optimization giving Updation of Direction vector with Change in velocity

3.3 Application of Particle Swarm Optimization

One Of the reasons that PSO is attractive is that there are few parameters to adjust. PSO has been used for approaches that can be used across a wide range of applications , as well as for specific application focused on a specific requirement.

Some of the applications for PSO are:

i. Evolving neural networks- evolving not only network weights but network structure also.
ii. Diagnosis of diseases like Parkinson disease.
iii. Metal removal operation in manufacturing environment
iv. For reactive power and voltage control
v. For monitoring of machines / equipments.

3.4 Variants of Particle swarm optimization

3.4.1 **Adaptive Particle Swarm Optimization**

Adaptive particle swarm optimization is a variant of basic particle swarm optimization

Adaptive PSO[9] is modification of PSO which at each stage eliminates the invaluable or less valuable particles , which on evolution have have undergone undesired process and have lost both local as well as global search capability and will not be able to provide optimal solution in later stages

The fitness of each particle is measured based on a Fitness function Fi.

Where

Fi is the fitness of ith particle

F_{gbest} is the fitness of gbest

.Δ Fi = $f(F_i, F_{gbest})$,where f(x) is a error function is calculated.

A ϵ is a predefined critical constant according to the precision requirement, T_c is the count constant

A replace function replaces all those particles in each iteration which fall below this constant $|\Delta Fi| < \epsilon$ thus eliminating all the inactive particles.

3.4.2 **Dynamic Particle Swarm Optimization**

They are called "dynamic" because the parameters c1 and c2 are updated during the convergence process

Dynamic Particle Swarm Optimization the cognitive and social behaviour are updated over iterations to focus on local as well as global best updation with iteration.

There are 2 algorithms, DPSO-1 and DPSO-2, there is an initial value for c_1 and c_2 :c_1^0 and c_2^0. These values are updated according to the rate of improvement of the population best fitness value along the interactions.

In these algorithms, Change in C1 and C2 is determined by the number of iteration currently in , rate of change for first 25% iterations is more than for next 25% of iteration and is decreased subsequently over iteration after every 25% of iterations. The reason behind decrease in change is initially more global value are to be searched and over the iteration near optimum solutions are produced and hence more local search is required.

Algorithm :

If the best fitness value remains the same during t interactions, c_1 and c_2 decrease by

- For first 25 % of iterations - s-35 % ie. c1 = c1(1-0.35s) and c_2=c_2(1-0.35s), where s - 1 is the number of times that c_1 e c_2 were already updated. These are minimum values for c_1 and c_2, defined as c_1^0(1-0.1p) and c_2^0(1-0.1p), respectively, where p is a positive integer constant.
- For next 25 % of iterations - s-25 % ie. c1 = c1(1-0.25s) and c_2=c_2(1-0.25s),
- For next 25 % of iterations - s-20 % ie. c1 = c1(1-0.20s) and c_2=c_2(1-0.20s),
- For final 25 % of iterations - s-10 % ie. c1 = c1(1-0.1s) and c_2=c_2(1-0.1s),

In DPSO-2, besides c_1 and c_2 decreasing mechanism, there is also an increasing rule:

- For first 25 % of iterations - s+35 % ie. c1 = c1(1+0.35s) and c_2=c_2(1+0.35s), where s - 1 is the number of times that c_1 e c_2 were already updated. These are minimum values for c_1 and c_2, defined as c_1^0(1+0.1p) and c_2^0(1+0.1p), respectively, where p is a positive integer constant.
- For next 25 % of iterations - s+25 % ie. c1 = c1(1+0.25s) and c_2=c_2(1+0.25s),
- For next 25 % of iterations - s+20 % ie. c1 = c1(1+0.20s) and c_2=c_2(1+0.20s),
- For final 25 % of iterations - s+10 % ie. c1 = c1(1+0.1s) and c_2=c_2(1+0.1s),

Chapter 4: Modeling

4.1 Assumptions

- It is assumed that all sensor nodes are homogeneous having same range
- Sensors are stationery.
- Sensing coverage and communication coverage of each node are assumed to have a spherical shape without any irregularity.
- The design variables are 3D coordinates of the sensor nodes, {(x1, y1,z1), ($x2$, $y2$,$z2$)……}.
- A 3 dimensional particle i represents x and y and z coordinates of all n sensors: Such that Xi = {x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,……xn,yn,zn} ,n= number of sensors
- f(Xi,Ri) = Volume coverage in 3D terrain by i th particle represented by Xi

4.2 **Volume Coverage**

We have considered a rectangular surface of dimension (M,L,B), where L,B & M are length , breadth and height of the Surface.

This rectangular surface is assumed to be completely covering the given terrain or region of interest . The rectangular surface represented by matrix data satucture . GridCoveredStausMatrix is having M slices of dimension (L,B). This rectangular surface is assumed to be divide into total (M*L*B) small grids. Here any cell of GridCoveredStausMatrix(m,i,j) matrix , stores the coverage status of (i,j) th point of m th Height Slice of this rectangular surface . GridCoveredStausMatrix is made of M slice. Height of any (i,j) th point of m th slice is assumed to be same. If the value of GridCoveredStausMatrix(m,i,j) is 1, then the (i,j) th grid region of the m th slice (Height) will be assumed to be covered by any of the sensors.

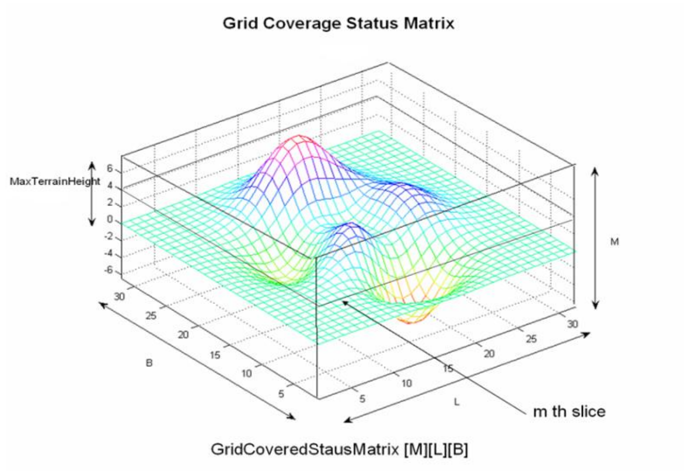

Figure 4.1 :Surface of MxLxB dimension , sliced in M layers of LxB dimension each

Initially no cells / grids are covered therefore Initialize all Grids of GridCoveredStausMatrix (m,i,j) with zero.

4.2.1 **To cover the maximum volume the problem reduces to**

Maximize f(Xi,Ri) - to miximize the area covered by n no of sensors

Where Xi = {x1,y1,z1,x2,y2,z2,x3,y3,z3,x4,y4,z4,……xn,yn,zn}

Ri= range of the i th sensor

f(Xi,Ri)= Coverage(Xi,Ri)

$$C = \frac{\bigcup_{i=1,\ldots,N} V_i}{V}$$

(Total volume coverage calculation)

where

Vi is the volume covered by the *ith* node;

N is the total number of nodes;

V stands for the volume of the Region of interest (ROI)

For coverage calculation Monte Carlo Method is used

Steps for coverage calculation

1. In order to prevent recalculating the overlapped area, the coverage here is calculated using Monte Carlo method by creating a uniform grid in the Region of interest.
2. All the grid points being located in the sensing area are labelled 1 otherwise 0, depending on the Euclidean distance between each grid point and the sensor node and also the terrain height is less than the height of grid point (for this Line of sight function from sensor centre to grid point is used).
3. Then the coverage can be approximated by the ratio of the summation of ones to the total number of the grid points.

PsuedoCode for calculation of coverage of volume

Estimation of Volume to be covered

```
ctr1=0;
voltobecovered=0;
for m=1:1:2*HeightVar +1
  for i=1:1:L
    for j=1:1:B
      HeightMatrix(m,i,j)=-HeightVar+ctr1;
      if ( HeightMatrix(m,i,j) < Z(i,j) )
        VolGridStatusMatrix(m,i,j)=-1 ;% below ground space (cannot  be covered )
      else
        VolGridStatusMatrix(m,i,j)=0 ; % Open space to be covered
        voltobecovered=voltobecovered+1;
      end
```

```
        end
      end
      ctr1=ctr1+1;
    end
    % for calculating Volume
    %Total volume to be covered
     voltobecovered;
```

4.2.2 Coverage by a single sensor

For the coverage calculation by single sensor , it assumed that sensing range of sensor is in spherical shape.

Thus , for calculation of 3-d region within sensing range of a sensor we can take into consideration the range of sensor , and calculate for every angle thetha and azimuthal for each angle . The following figure demonstrates the concept.

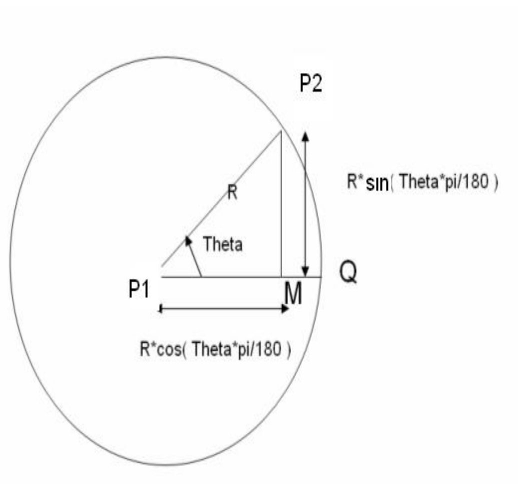

Figure 4.2 Diagram showing Azimuthal calculation of are covered by a sensor.

Psuedocode for calculation of coverage of volume by one sensor and subsequently one particle

Coverage (Xi,R) by one PSO particle

This function gives total volume covered by the solution vector of ith particle

```
Coverage(Xi,R)
  For  k=1:2:size(Xi)
   {
      extracting x location of k th sensor nodes in i th PSO particle
      x1=Xi(k)
      extracting y location of k th sensor nodes in i th PSO particle
      y1=Xi(k+1)
      z1=Z(x1,y1)
      creating a k th sensor node location and placing the sensor at height z1 which is the actual
height of the terrain
      P1 =[x1,y1,z1];
     ctr=1;

 for j=start_azim:g:end_azim
      compute  height at azim angle j (degree)
      height= R*sin(j*pi/180);
     effective sphere radii at azim angle j (degree)
      RR= R*cos(j*pi/180);
      for i=start_angle:g:end_angle
        complete location of extreme point P2 oriented at angle i degree (see fig 1)
```

```
x2= x1+RR*cos(i*pi/180);
y2= y1+RR*sin(i*pi/180);
z2=z1+ height;
 P2= [ x2,y2,z2];
  X1=0 ;        Y1=0;        Z1=0;
```

compute intermediate points [X1 Y1 Z1} in between P1 node centre to P2 extreme points

```
 [X1 Y1 Z1  ]=Points_On3d_Line(P1,P2,1);
 [Los_Flag TempVolGridStatusMatrix]  = LOS_OR_NOT1(X1 ,Y1
 ,Z1,Z,Peaksize,ctr,TempVolGridStatusMatrix,HeightMatrix,HeightVar) ;

  end  %i
 ctr=ctr+1;
 end %j
end //  for k th sensor node
```

To check the distance between two points in 3-Dimension , and to find out whether its an elevated path or a regular path from one point to another a function is used , which is required for calculation of volume covered by a sensor o line of sight function is used. It provides intermediate points between two points P1 and P2

Psuedocode for line of sight function

<u>Estimation of intermediate points from sensor center P1 to farthest range points P2</u>

function[X1 ,Y1, Z1] gives intermediate points between P1 to p2 at step of g

```
function[X1 ,Y1, Z1  ]=Points_On3d_Line(P1,P2,g)
d=sqrt((P1(1)-P2(1))^2+(P1(2)-P2(2))^2+(P1(3)-P2(3))^2);
```

```
r=0:g:d;
dcs=[];
for i=1:3
  dcs(i)=(P1(i)-P2(i))/d;
end
X1=P1(1)-r.*dcs(1);
Y1=P1(2)-r.*dcs(2);
Z1=P1(3)-r.*dcs(3);
```

4.3 Calculation for energy dissipation

The system is required to transmit data to a SYNC node outside the network to transmit data , since all the nodes are connected to each other (through direct and indirect)hence data travel though nodes till it reaches upto SYNC.

Since there would be intermediate nodes acting as bridges/transmitter and transmitting their data as well , hence energy efficiency is considered to determine the longevity of network and provide

4.3.1 **Assumptions , calculations and energy function**

An energy Function is is required for the calculation of dissipation of energy. Following points are taken into consideration to formulate Enegy dissipation function

- Sensors in WSN have tasks like data acquisition, processing and transmitting , all require energy , and thus some amount of energy is dissipated on each action.
- Some of the energy required is constant which is sensor dependent , the rest depends on the distance the data is to be transmitted and the amount of data to be transmitted.

  The energy dissipated is given by :

$$\mathbf{E = \alpha_1 . r_d + \alpha_2 . d_{i,j}{}^2 . r_d} \quad [10]$$

  Where

E – Total energy dissipated

α_1 – electronic energy dissipated in transmitting one bit of data.

r_d – data rate

$d_{i,j}$ – it is the distance between node and can be calculated using Euclidian formula in 3d

Using this energy and applying on PSO algorithm an optimized solution is found

Procedure to find total energy dissipated in a particle depends on the approach (Relay or Clustered) followed and are given in detail in subsequent chapters. This describe the common assumptions and calculation common to all procedures.

Chapter 5: Relay Approach

5.1 Introduction

Initial implementation of this research considered relay approach to find the the maximum volume coverage and energy efficiency.

In relay all the nodes are placed such that they form a relay structure , i.e. the nodes are placed such that they can transmit message from one node to other to the next till it reached SYNC node.

5.2 Procedure to form a relay structure

1. Initially , all the particles are randomly initialized , in such manner for n sensors, such that location for first node is found randomly and fixed , then within R communication range of this sensor position of next sensor is found and this is subsequently carried out for n sensors
2. After initialization , the particles are fed into PSO algorithm , over the iterations these nodes change their position to form a structure for maximum volume and energy efficiency coverage.
3. During Iterations , the nodes might break the relay structure , in order to keep the relay structure intact Relay check algorithm is used to check for relay
4. If any break found , a relay repair algorithm is applied to reconstruct the relay structure again on the broken nodes (Algorithm is discussed in detail in following sections)

5.3 Repair Algorithm

While randomizing and then updating the velocity and direction of each particle in PSO , some dimension in solution of a particle may violate minimum 2r constraint .In order to keep the solution given by each particle conferring to the constraint of maximum distance of 2r to maintain relay , a repair algorithm is used which works as follows.

Algorithm :

1) If the solution is found to be not suitable according to the constraints by applying relay check algorithm over each particle to check whether there there exist a relay in solution or not

2) A repair algorithm is applied over the solution to modify and adjust the solution. The disjkatra based repair algorithm takes each candidate Particle configuration (node location) and gives output as relay feasible / Not relay feasible. If the particle configuration is not relay feasible, the repair algorithm also returns the node index from where the relay need to be repaired

3) In repair algorithm the node violating the constraint are randomly moved over space a little and relay check algorithm is applied over each new configuration till valid configuration is found.

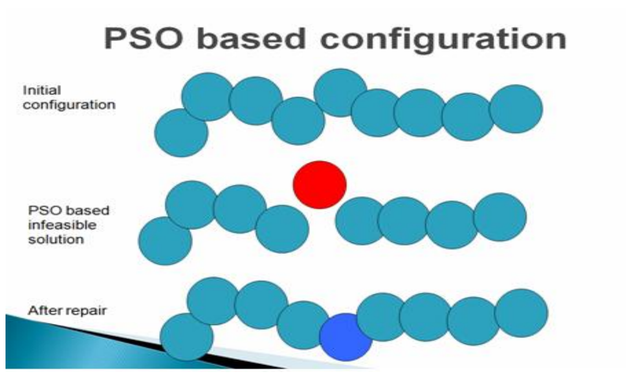

Figure 5.1 : Diagram for Repair Algorithm

5.4 **Procedure for calculation energy Dissipation.**

Energy efficiency function

Sensors in WSN have tasks like data acquisition, processing and transmitting , all require energy , and thus some amount of energy is dissipated on each action.

Some of the energy required is constant which is sensor dependent , the rest depends on the distance the data is to be transmitted and the amount of data to be transmitted.

The energy dissipated is given by :

$$\mathbf{E = \alpha_{1}.r_d + \alpha_{2}.d_{i,j}^{\,2}.\, r_d}$$

Where

E – Total energy dissipated

α_1 – electronic energy dissipated in transmitting one bit of data.

r_d – data rate

$d_{i,j}$ – it is the distance between node and can be calculated using Euclidian formula in 3d

Using this energy and applying on PSO algorithm an optimized solution is found

Procedure to find Energy dissipation

- At each iteration of PSO after applying energy function, Using dijsktra algorithm a network is formed to create a shortest distance tree, which gives shortest distance of every node from SINK node , thorough intermediate nodes.
- After creating the network a Matrix $d_{i,j}$ is created which gives the distance between two nodes i and j. A complete route given by Dijkastra algorithm from node i to j is of the form of node i, m, n,…j., where m, n are intermediately nodes. The value $d_{i,m}$ $d_{m,n}$, $d_{n,j}$ represent distance from node i to m,m to n, n to j respectively. This value value $d_{i,m}$ $d_{m,n}$, $d_{n,j}$ given distance matrix (d) is used to estimate total energy dissipated for each particle using above mentioned energy function
- After creating the network , a $M_{i,j}$ matrix is created which keeps track of how many times

data is passed from i,j path (i.e from node i to node j) . data will pass through node i to j , when node i transmits its data and node j happens to be next in network towards path of SYNC node or node i transmits data of previous node in network to node j.

- Matrix $M_{i,j}$ is calculated by traversing path of each node to SINK node in the network formed by using djistkra algorithm. This matrix gives the amount of data being transmitted over each path . (It is assumed that nodes at same time transmit data to SYNC node).
- Using this matrix after every iteration and creation of matrix, total energy dissipated is calculated for each particle using above mentioned energy function.

5.5 Simulation Results

Assumptions and Parameters

In the particular scenario we have considered a terrain of 32 * 32 meter. Maximum Terrain Height considered was 4 .In this terrain taken 32 number of sensors are to be deployed optimally . sensor range has been taken as 4 meter.

Table 1. Fundamental parameters of WSN.

Parameter	**Value**
Sensing field dimension *LxL* :	*32 meter * 32 meter*
Stationary node number *n:*	*32*
Sink node coordinates:	(4,4)
Sensing radius *Ra :*	3m
start_angle :	0 degree
end_angle:	180 deg
Azim:	35 degree
azim_span:	70 degree

start_azim:	0 degree
end_azim :	70 degree

In the energy consumption model, we **set α1 = 50*nJ*/*bit*** and **α2 = 100*pJ*/*bit*/*m2***

Table 2 :PSO Parameters

The inertia weight (w) is initially set as 1.0 and decreasing to 0.0 linearly over the iterations .

No of particles:	**20**
c1	**1.6**
c2=	**1.4.**
Vmin	**-5**
Vmax	**5**

5.5.1 Using Basic Particle Swarm Optimisation

Using Basic Swarm Particle Optimizations the , 20 particles are iterated over 100 iterations to generate the result , in each iteration each particle updates its local best and using local as well as local best updates its velocity and position subsequently , thus searching the maximum volume coverage with Energy Efficiency.

Total Volume Coverage

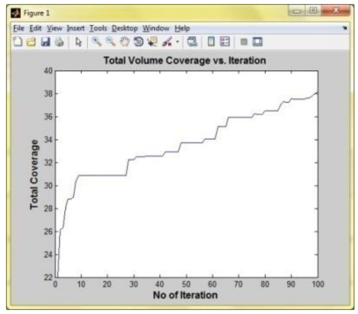

Figure 5.2.1 Total Volume Coverage over iteration for Basic PSO

Total Energy Dissipation

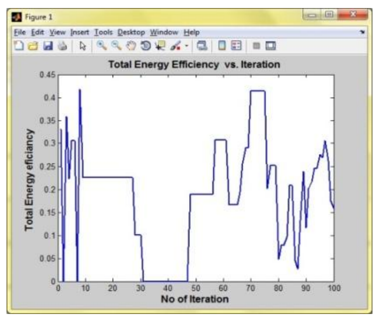

Figure 5.2.2 Total Energy Dissipation over iteration for Basic PSO

Total MOE(Measure Of Effectiveness)

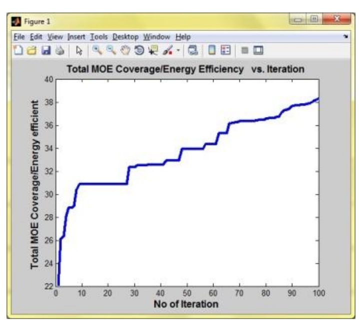

Figure 5.2.3 Total MOE over iteration for Basic PSO

Final 3-d represenataion of volume coverage

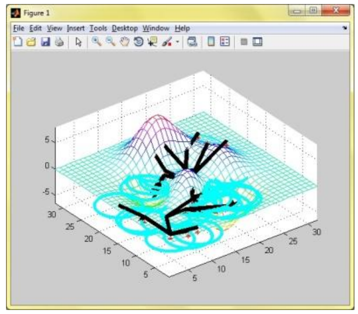

Figure 5.2.4 Final 3-D Volume Coverage for Basic PSO

Analysis :

Volume Coverage given by Basic PSO , increases from 22 % to 38-39% over 100 iteration , with an increase of around 16% from initial deployment.

Energy Dissipation , reduced from 0.35 to nearly 0.16 , thus taking into consideration energy constraints too.

5.5.2 **using Adaptive particle Swarm Optimization**

APSO Parameters

- For **APSO,** Δ *Fi* is set as a relative error function, which is
- (*Fgbest*- *Fi*)/MAX(ABS(*Fi*), ABS(*Fgbest*)), where ABS(*x*) gets the absolute value of *x*, MAX(*x*1, *x*2) gets the minimum value between *x*1 and *x*2.
- The critical constant ϵ is set as 10-4, and the count constant *Tc* is set as 3

Total Volume Coverage

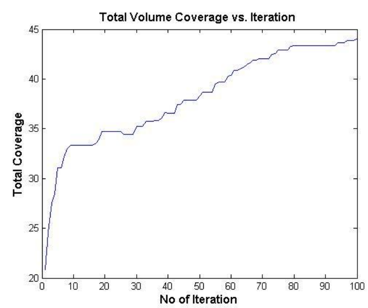

Figure 5.3.1 Total Volume Coverage over iteration for APSO

Total Energy Dissipation

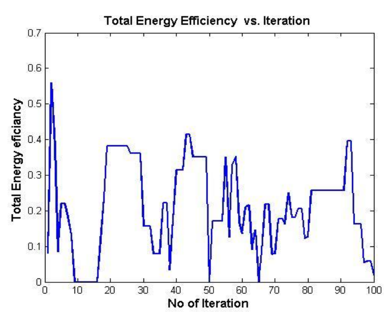

Figure 5.3.2 Total Energy Dissipation over iteration for APSO

Total MOE(Measure Of Effectiveness)

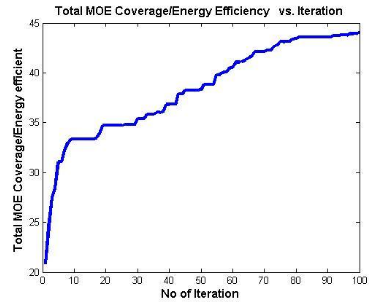

Figure 5.3.3 Total MOE over iteration for APSO

Final 3-d represenataion of volume coverage

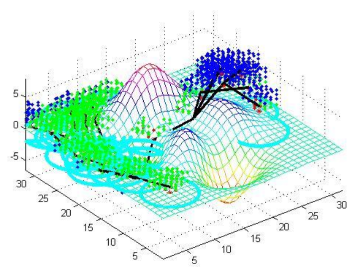

Figure 5.3.4 Final 3-D volume Coverage for APSO

Analysis :

Volume Coverage given by APSO , increases from 21 % to 44% over 100 iteration , with an increase of around 23% from initial deployment.

Energy Dissipation , reduced from 0.55 to nearly 0.05 , thus taking into consideration energy constraints too.

5.5.3 **using Dynamic particle Swarm Optimization**

For DPSO , C1 nad C2 are updated continuously over iterations.

code for updation of C1 and C2

```
 if ( max(T_LBest)<  GBest )
   it_ctr=it_ctr+1;
   it_ctr2=0;
 end

 if( it_ctr >=t_ctr)
   sol_mdified_dpso1=sol_mdified_dpso1+1;

   if((No_Iteration /MAX_ITERATION)<=0.25)
     c1=c1*(1-.35*sol_mdified_dpso2);
     c2=c2*(1-.35*sol_mdified_dpso2);
     end
      if((No_Iteration /MAX_ITERATION)>0.25 && (No_Iteration /MAX_ITERATION)<=0.50 )
     c1=c1*(1-.25*sol_mdified_dpso2);
     c2=c2*(1-.25*sol_mdified_dpso2);
      end
      if((No_Iteration /MAX_ITERATION)>0.50 && (No_Iteration /MAX_ITERATION)<=0.75 )
     c1=c1*(1-.2*sol_mdified_dpso2);
     c2=c2*(1-.2*sol_mdified_dpso2);
      end
      if((No_Iteration /MAX_ITERATION)>0.75)
     c1=c1*(1-.1*sol_mdified_dpso2);
     c2=c2*(1-.1*sol_mdified_dpso2);
     end
   it_ctr=0;
 end

 % if changes value of c1 , c2 goed below threshold,
```

```
if(c1<c11)
   c1=c11;
end

 if(c2<c22)
   c2=c22;
end

% end DPSO  1 code



% DPSO 2 code
  if ( max(T_LBest)>  GBest )
         it_ctr2=it_ctr2+1;
         it_ctr =0;
  end



 if( it_ctr2 >=t_ctr)



       it_ctr2=0;
       sol_mdified_dpso2=sol_mdified_dpso2+1;

       if((No_Iteration /MAX_ITERATION)<=0.25)
       c1=c1*(1+.35*sol_mdified_dpso2);
       c2=c2*(1+.35*sol_mdified_dpso2);
       end
        if((No_Iteration /MAX_ITERATION)>0.25 && (No_Iteration /MAX_ITERATION)<=0.50 )
```

```
        c1=c1*(1+.25*sol_mdified_dpso2);
        c2=c2*(1+.25*sol_mdified_dpso2);
         end
         if((No_Iteration /MAX_ITERATION)>0.50 && (No_Iteration /MAX_ITERATION)<=0.75 )
        c1=c1*(1+.2*sol_mdified_dpso2);
        c2=c2*(1+.2*sol_mdified_dpso2);
         end
         if((No_Iteration /MAX_ITERATION)>0.75)
        c1=c1*(1+.1*sol_mdified_dpso2);
        c2=c2*(1+.1*sol_mdified_dpso2);
        end

         if(c1>c31)
           c1=c31;
         end

         if(c2>c32)
           c2=c32;
         end




   end  %  if( it_ctr2 >=t_ctr)
```

Total Volume Coverage

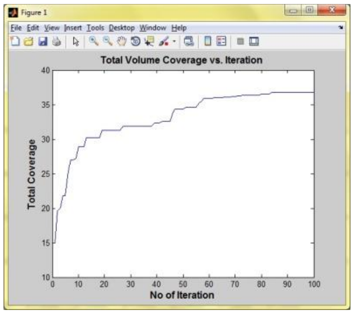

Figure 5.4.1 Total Volume Coverage over iteration for DPSO

Total Energy Dissipation

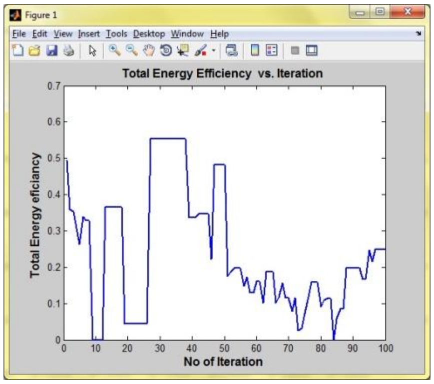

Figure 5.4.2 Total Energy Dissipation over iteration for DPSO

Total MOE(Measure Of Effectiveness)

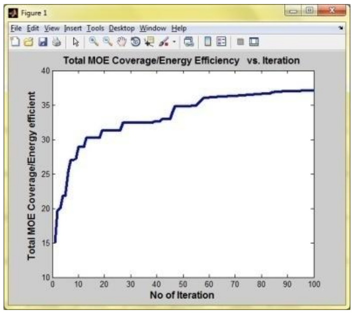

Figure 5.4.3 Total MOE over iteration for DPSO

Final 3-d represenataion of volume coverage

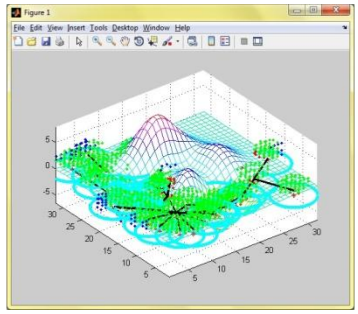

Figure 5.4.4 Final 3-D Volume Coverage for DPSO

Analysis :

Volume Coverage given by Basic PSO , increases from 15 % to 37% over 100 iteration , with an increase of around 22% from initial deployment.

Energy Dissipation , reduced from 0.5 to nearly 0.25 , thus taking into consideration energy constraints too.

For Relay Approach Volume coverage is best given by APSO and DPSO , but APSO gives overall best result as Energy Efficiency in APSO soln is better. (reduction of 0.5).Thus , APSO is preferable here.

Chapter 6: Clustered Approach

6.1 Introduction

In this approach , a clustered network is formed, where in there is a backbone , few nodes on backbone act as cluster heads , these cluster head are responsible for data aggregation and to the transmitting the data tp SYNC nodes.

The transmission through cluster head takes place in relay form to the SYNC node.

6.2 Procedure for volume coverage

1. Initially , the backbone is constructed , total no of sensors are known , these sensors are divided into two parts , one part which will form the backbone and other which will be cluster nodes . for this application a ratio of 2:3(backbone nodes : cluster nodes) is used.
2. For creating backbone , same relay approach is used , which forms a structure for the backbone , for this an initial PSO algorithm is used. Backbone remains fixed through cluster deployment.
3. For cluster nodes , cluster nodes are randomly deployed as per ratio around cluster heads in backbone , initially a random deployment approach is used , later this random deployment is fed into another PSO algorithm which calculates the optimal solution.
4. For this application , every cluster contains 4 nodes including cluster heads , in backbone cluster are formed at every alternate nodes . Thus giving rise to a ratio of 2:3 (although this ratio can be adjusted to any ratio suitable according to the application).
5. Intermediate nodes of backbone which are not used as cluster nodes are responsible for sensing the region within their range and transmit data from previous nodes in the Backbone structure to the Next node in relay or the SYNC node.

6.3 Procedure for calculation of energy Dissipation

Energy Function

Function used to calculate energy Dissipation is same as in Relay Approach (Section 5.4)

The energy dissipated is given by :

$$E = \alpha_1 . r_d + \alpha_2 . d_{i,j}{}^2 . r_d$$

Procedure to calculate Energy Dissipation

- For Calculation of energy Dissipated we have to calculate , the energy dissipated at each backbone node for transmitting the message to the next in relay , this is done in the same procedure using djikatra algorithm and finding shortest path to SYNC node for each node as in relay approach (see section
- Message transferred by every cluster node would be equal to messages transferred by previous cluster node in relay plus the node in cluster nodes associated with this node.
- Each cluster node has to transfer only message to its cluster node (cluster_MOE) , thus for the same purpose a distance matrix cluster_nodes_distance(i,BB_node_no,C_node_no) is maintained which keeps distance of each cluster node from its cluster head. Thus energy dissipated for cluster nodes can be found.
- Since each node in backbone has to transmit messages from its cluster members , its messages and message from previous node in relay of backbone . Thus the total messages being transmitted by nodes have to be determined .
- A matrix nodes_assocaited(i,BB_node_no) is maintained for each cluster head to keep record of no of cluster nodes associated with this node , thus giving no of messages received from cluster.
- Using Djikastra algorithm message transferred for each node is calculated ,
  Total packets transferred by a node is given by , msgpckttransfered(from previous node)+nodesassciated + 1 , using this information with distance information in Energy Function , total energy Dissipated can be found (relay MOE)
  Thus total EMOE = realy_MOE+cluster_MOE

Psuedocode for Energy Calcualtion

```
Cluster_MOE
{     Cluster_EMOE  =0;
    For i=0 to cluster nodes
 cluster_EMOE = cluster_EMOE + 1*(alpha1* distance + alpha2*distance^2)

   return cluster_EMOE;
}



Relay MOE
{    relay_EMOE=0;
      For i=0 to backbone_nods
{
     msgpackt msgpacktransfered = msgpacktransfered + nodes_assocaited(1,path1(j)) + 1;


    Relay_EMOE = cluster_EMOE + msgpacktransfered*(alpha1* distance alpha2*distance^2)
}

Return relay_EMOE
}
Toatal_energy = cluster_EMOE+ Relay_EMOE
```

6.4 Simulation Results

Assumptions and Parameters

In the particular scenario we have considered a terrain of 32 * 32 meter. Maximum Terrain Height considered was 4 .In this terrain taken 32 number of sensors are to be deployed optimally . sensor range has been taken as 4 meter.

Table 3. Fundamental parameters of WSN.

Parameter	**Value**
Sensing field dimension *LxL* :	*32 meter * 32 meter*
Stationary node number *n:*	*32*
Sink node coordinates:	(4,4)
Sensing radius *Ra :*	3m
start_angle :	0 degree
end_angle:	180 deg
Azim:	35 degree
azim_span:	70 degree
start_azim:	0 degree
end_azim :	70 degree

In the energy consumption model, we **set α1 = 50*nJ/bit*** and **α2 = 100*pJ/bit/m2***

Table 4 : PSO Parameters

The inertia weight (w) is initially set as 1.0 and decreasing to 0.0 linearly over the iterations .

No of particles:	**20**

c1	**1.6**
c2=	**1.4.**
Vmin	**-5**
Vmax	**5**

6.4.1 Using Basic Particle Swarm Optimisation

Using Basic Swarm Particle Optimizations the , 20 particles are iterated over 100 iterations to generate the result , in each iteration each particle updates its local best and using local as well as local best updates its velocity and position subsequently , thus searching the maximum volume coverage with Energy Efficiency.

Total Volume Coverage

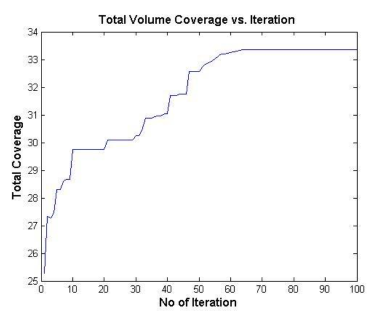

Figure 6.1.1 Total Volume Coverage over iteration for Basic PSO

Total Energy Dissipation

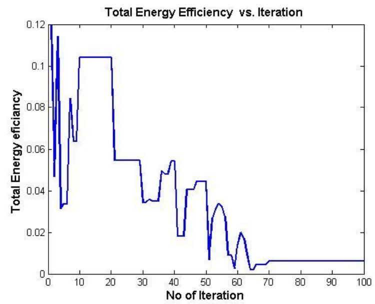

Figure 6.1.2 Total Energy Dissipation over iteration for Basic PSO

Total MOE(Measure Of Effectiveness)

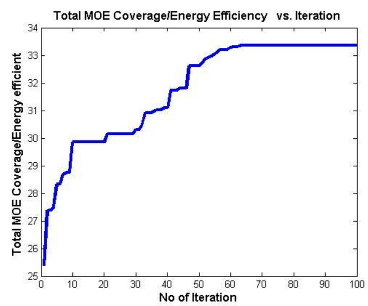

Figure 6.1.3 Total MOE over iteration for Basic PSO

Final 3-d represenataion of volume coverage

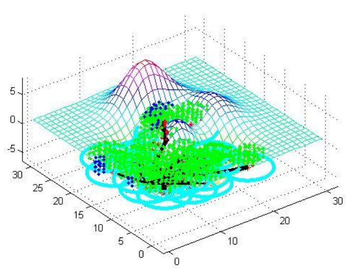

Figure 6.1.4 Final 3-D Volume Coverage for Basic PSO

Analysis :

Volume Coverage given by Basic PSO , increases from 25.5 % to 33.5% over 100 iteration , with an increase of around 8% from initial deployment.

Energy Dissipation , reduced from 0.12 to nearly 0.01 , thus taking into consideration energy constraints too.

6.4.2 **using Adaptive particle Swarm Optimization**

APSO Parameters

- For **APSO,** Δ *Fi* is set as a relative error function, which is
- (*Fgbest*- *Fi*)/MAX(ABS(*Fi*), ABS(*Fgbest*)), where ABS(*x*) gets the absolute value of *x*, MAX(*x*1, *x*2) gets the minimum value between *x*1 and *x*2.
- The critical constant ϵ is set as 10-4, and the count constant *Tc* is set as 3

Total Volume Coverage

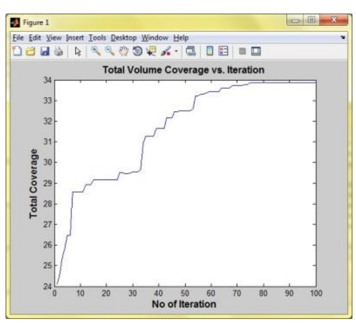

Figure 6.2.1 Total Volume Coverage over iteration for APSO

Total Energy Dissipation

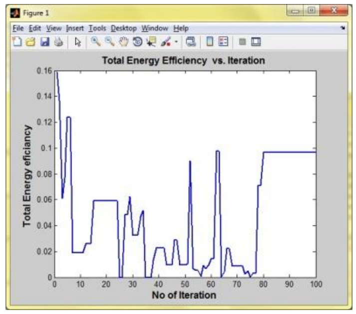

Figure 6.2.2 Total Energy Dissipation over iteration for APSO

Total MOE(Measure Of Effectiveness)

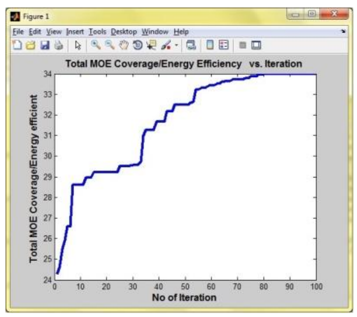

Figure 6.2.3 Total MOE over iteration for APSO

Final 3-d represenataion of volume coverage

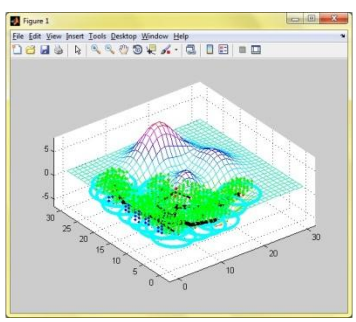

Figure 6.2.4 Final 3-D Volume Coverage for APSO

Analysis :

Volume Coverage given by APSO , increases from 24 % to 34% over 100 iteration , with an increase of around 10% from initial deployment.

Energy Dissipation , reduced from 0.16 to nearly 0.09 , thus taking into consideration energy constraints too.

6.4.3 **using Dynamic particle Swarm Optimization**

For DPSO , C1 nad C2 are updated continuously over iterations.

Total Volume Coverage

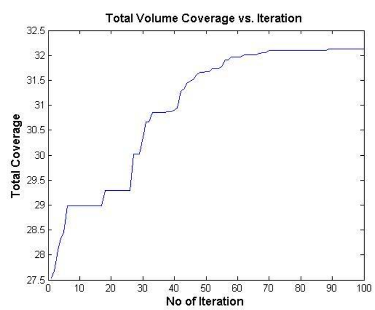

Figure 6.3.1 Total Volume Coverage over iteration for DPSO

Total Energy Dissipation

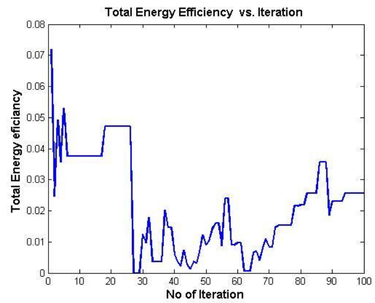

Figure 6.3.2 Total Energy Dissipation over iteration for DPSO

Total MOE(Measure Of Effectiveness)

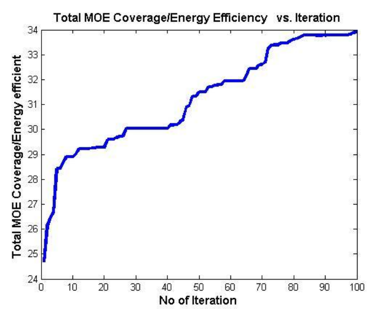

Figure 6.3.3 Total MOE over iteration for DPSO

Final 3-d represenataion of volume coverage

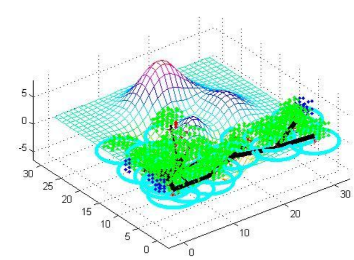

Figure 6.3.4 Final 3-D Volume Coverage for DPSO

Analysis :

Volume Coverage given by DPSO , increases from 27.5 % to 32.5% over 100 iteration , with an increase of around 5% from initial deployment.

Energy Dissipation , reduced from 0.07 to nearly 0.025 , thus taking into consideration energy constraints too.

In Clustered Approach , APSO gives the Best volume coverage as Volume increase is around 10% , for energy efficiency Basic PSO and DPSO gives better result but not very effective Volume Coverage with volume increase only limited to 7-8%.
If energy efficiency is of utmost importance then Basic PSO is the correct approach, Else APSO gives the Best possible Solution.

Chapter 7: Comparative Study

The following chapter gives Comparison between the simulation Results

7.1 Relay vs Clustered approach

Approach	Basic PSO			APSO			DPSO		
	volume	energy	MOE	volume	energy	MOE	volume	energy	MOE
Relay Approach	22to 38 - 39% , Volume increase - 16%	0.35 to 0.16	40% Approx.	21 % to 44% Volume increase – 23%	0.55 to 0.05	45 % Approx.	15% to 37% volume increase- 22%	0.5 to 0.25	37 % approx.
Clustered Approach	25.5 to 33.5% , Volume increase - 8%	0.12 to 0.01	34% Approx.	24 % to 34% Volume increase – 10%	0.16 to 0.09	34% Approx.	27.5 % to 32.5% volume increase- 5%	0.07 to 0.025	33% approx.

Table 5 : Comparison between relay and Clustered approach

Analysis:

The Above Table Summarizes Results of Relay and Cluster Approach , It can be seen that Relay Approach gives better volume coverage for same number of sensors , as overlapping between region Covered by sensors is minimum ,maximum volume covered in relay approach is given by APSO i.e. 44 % and in cluster is 34 % , but as expected there is a significant decrease in Energy Dissipation in cluster Approach. Energy Dissipation reduces to as low as 0.01 in comparison to average energy dissipation of relay approach 0.1 to 0.2. Thus , this assert that clustered approach gives the best solution when Energy Efficiency is of utmost importance in comparison to volume coverage and from the results it can be seen that with not much significant decrease in volume coverage it provide a solution with significant decrease in Energy Dissipation. Thus , leading to energy efficient clustered deployment of wireless sensor networks.

7.2 PSO vs APSO vs DPSO

Algorithm	Relay Structure			Cluster Structure		
Basic PSO	Volume increase	Final Energy	MOE	Volume increase	Final Energy	MOE
	16%	0.16	40%	8%	0.01	34%
APSO	23%	0.05	45%	5%	0.09	34%
DPSO	22%	0.25	37%	10%	0.025	33%

Table 6: comparison between PSO vs APSO vs DPSO

Analysis:

Basic PSO has given a decent increase in volume coverage in both relay and cluster approach and has maintained good energy efficiency overall.

APSO has shown best increase in volume in both relay and cluster with very good energy efficiency

DPSO has shown good increase in volume but not more than APSO.

APSO has proven to be most efficient algorithm for search of optimum solution

Thus , it is evident from the table that APSO is the best as it replaces all the particles over iteration which does not show much growth , followed by DPSO and PSO respectively.

Chapter 8: Conclusion and Future Work

8.1 Conclusion

Wireless sensor networks are technology of future , Wireless sensors pose some challenges like very less power backup and limited sensing range , thus energy efficient deployment of sensors is required to increase the longevity and cover maximum volume possible through sensors.

With limited number of sensors provided , the problem is to find maximum volume covered , for this to find the optimal solution a search algorithm was required and PSO is used in this application for finding the optimal solution.

Two approaches were used , relay approach and cluster approach , both the approach provided significant solutions but cluster approach showed best result for energy efficiency as expected and provided energy efficient cluster deployment of wireless sensor nodes.

Finally there was a comparison between three Variants of PSO , Basic PSO , APSO and DPSO , Results show that APSO gave best results for relay as well as clustered approaches.

Thus , an optimum solution for energy efficient deployment of Wireless sensor nodes could be achieved by using APSO with clustered approach.

8.2 Future Work

For Future Work , this work can be extended , to cover maximum Volume of critical regions within the whole Monitored Area , by identifying critical areas and giving them priotiy over other regions .

Clustered Approach using heterogeneous nodes or sensors where cluster heads having more transmitting range than cluster nodes can be used to reduce the dependency on relay back bone.

Self Clustering Algorithm can be introduced where in nodes are deployed randomly and over iteration they form self cluster , considering various parameters thus giving self organizing network .

References

[1]kai Xie, Zhengbin Yang, Zhitao Huang, Yiyu Zhou, LEO Space based Radar Constellation Design Using A Genetic Algorithm, College of Electronic Science & Engineering,National University of Defense Technology, Changsa 410073, China

[2] , Yulai Suan ,A genetic – Algorithm based Mobile Sensor , Network Deployment Algorithm., EE382C : Embedded Software Systems, department of Electrical & computer Engineering The University of Texas , Austin.

[3] Damien B. Jourdan, Oliver L de Weck Layout Optimization for a Wireless Sensor Network using a multi – Objective Genetic Algorithm.,., Dept of Aeronautics and Astronautics, Massachusetts Institute of wireless sensor networks" Technology 77, Massachusetts I Avenue, Cambridge, MA 02139, USA

[4] Xue Wang , Jun-jie Ma , Sheng Wang , Dao-wei Bi ,Sensors 2007 ,"Distributed particle swarm optimization and simulated annealing for energy-efficient coverage in wireless sensor networks"

[5] http://en.wikipedia.org/wiki/Wireless_sensor_network

[6]http://en.wikipedia.org/wiki/Particle_swarm_optimization

[7]http://www.swarmintelligence.org/s

[8] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". Proceedings of IEEE International Conference on Neural Networks. IV. pp. 1942–1948.

[9]Adaptive Particle Swarm Optimization, Zhi-hui Zhan and Jun Zhang ,Department of Computer Science, Sun Yat-sen University, China,junzhang@ieee.org

[10] sECTION 2.2 Distributed Particle Swarm Optimization and Simulated Annealing for Energy-efficient Coverage in Wireless Sensor Networks Xue Wang *, Jun-Jie Ma , Sheng Wang and Dao-Wei Bi

[11]J. H. Holland,Adaptation in natural and artificial systems, Ann Arbor. , USA, The University of Michigan Press, 1975.

[12]Holland, J. H ,Genetic Algorithms , Sceintific American, 66-72, July 1992

[13]Davis, L Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York 1991

[14]Goldberg, D. EGenetic Algorithm in search, Optimization and Machine Learning. Addisson Wesley, New York, 1988

[15]Yuhui Shi,Particle Swarm Optimization, Electronic Data Systems, Inc. Kokomo, IN 46902, USA

[16]Xiao-Feng Xie, Wen-Jun Zhang, Zhi-Lian Yang," Adaptive Particle Swarm Optimization on Individual Level" , Institute of Microelectronics, Tsinghua University, Beijing 100084, P. R. China ,International Conference on Signal Processing (ICSP), Beijing, China, 2002: 1215-1218

[17]Wireless Sensor Networks: To Cluster or Not To Cluster? N. Vlajic and D. Xia Department of Computer Science and Engineering York University,Toronto, Canada vlajic@cs.yorku.ca , dxia@cs.yorku.c

[18]A Clustering Routing Algorithm Based on Adaptive PSO in WSNs , Jingming Cai ,jingming08@gmail.com AND Jifeng Sun , College of Electronics and Information ,South China University of Technology ,Guangzhou, China, 510640,ecjfsun@scut.edu.cn

[19]A survey on clustering algorithms for wireless sensor networks , Ameer Ahmed Abbasi a,*, Mohamed Younis b , a Department of Computing, Al-Hussan Institute of Management and Computer Science, Dammam 31411, Saudi Arabia , b- Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD 21250, USA

[20]An Intelligent Energy Efficient Clustering in Wireless Sensor Networks Jamshid Shanbehzadeh, Saeed Mehrjoo, Abdolhossein Sarrafzadeh

[21]A Survey of Clustering Algorithms for Wireless Sensor Networks ,D. J. Dechene, A. El Jardali, M. Luccini, and A. Sauer. Department of Electrical and Computer Engineering , The University Of Western Ontario , London, Ontario, Canada {ddechene, aeljarda, mluccini, asauer2}@uwo.ca