ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)

SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING

Optimal Wireless Sensor Networks Deployment in 3-Dimensional Terrains using Hybrid Population Based Algorithm

By

Yiwab Enyew Tessema

February, 2014

Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ADDIS ABABA INSTITUTE OF TECHNOLOGY (AAiT)

SCHOOL OF ELECTRICAL & COMPUTER ENGINEERING

Optimal Wireless Sensor Networks Deployment in 3-Dimensional Terrains using Hybrid Population Based Algorithm

By

Yiwab Enyew Tessema

Advisor

Dr. Yalemzewud Negash

Co-advisor

Ato Menore Tekeba

A thesis submitted to the school of Graduate studies of Addis Ababa University in partial fulfillment of the requirements for the degree of Masters of Science in Communication Engineering

February, 2014, Addis Ababa, Ethiopia

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

Optimal Wireless Sensor Networks Deployment in 3-Dimensional Terrains using Hybrid Population Based Algorithm

By

Yiwab Enyew Tessema

ADDIS ABABA INSTITUTE OF TECHNOLOGY

APPROVAL BY BOARD OF EXAMINERS

Prof. H.Y. (Hayal) Kwon	_____
Chairman, Dept. of Graduate Committee	Signature
Dr. Yalemzewud Negash	_____
Advisor	Signature
_____	_____
Internal Examiner	Signature
_____	_____
External Examiner	Signature

Declaration

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been fully acknowledged.

Yiwab Enyew Tessema _____

Name signature

Place: Addis Ababa

_____

Date of submission

This thesis has been submitted for examination with my approval as a university advisor.

Dr.Yalemzewud Negash _____

Advisor's name Signature

Acknowledgments

I owe my deepest gratitude to my advisor, Dr. Yalemzeud Negash and for my Co-advisor Ato Menore Tekeba, for their sincere support, unlimited encouragement, and meaningful guidance during the whole process from initial research to final thesis report. My thesis would not be completed well without my advisors support. They have always been available and willing to help me. Besides of God! I would also like to thank my family and my husband for their pure love and understanding.

Table of Contents

Table of Figures

Table Entries

Acronyms and Abbreviations

2-D, 3-D: 2 Dimensional plane and 3 Dimensional Volume or Surface (terrain) respectively

APSO: Adaptive Particle Swarm Optimization

GA: Genetic Algorithm

HA: Hybrid Algorithm

HPSOGA: Hybrid of Particle Swarm Optimization and Genetic Algorithm

LOS: Line of Sight

NLOS: Non-Line of Sight

PSO: Particle Swarm Optimization

QoC: Quality of Coverage

RoI: Region of Interest

SD: Sensor Deployment

WSNs: Wireless Sensor Networks

Abstract

Wireless sensor networks (WSNs) are composed of cooperating sensor nodes that can perceive the environment to monitor physical phenomena and events of interest. Sensor deployment is a fundamental issue in a WSNs to maximize coverage and quality of service with limited number of sensor nodes. In order to maximize area coverage, sensors need to be placed in a position such that the sensing capability of the network reach at high quality. Coverage is one of the main problems in WSNs deployment. Previous research works on sensor deployment mainly focused on Two Dimensional (2D) plane or in Three Dimensional (3D) volume coverage. But now, these studies on sensor deployment extended to 3D surfaces or terrain, to achieve the highest overall sensing quality. In our thesis, we worked to develop an optimal WSNs deployment on 3D surfaces to maximize area coverage under constrained number of nodes.

Researchers have used different methods and algorithms to make sensor deployment. Population-based optimization algorithms find near-optimal solutions to the difficult optimization inspired by natural probabilistic evolution. In our research work, Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) are selected to form Hybrid Algorithm (HA) to find optimal locations of sensors based on a fitness function. We have selected the two algorithms to exploit the best features of the algorithms in combination.

We have used two typical surfaces, rough and smooth, to compare the results of the GA, PSO and HA in the optimal deployment of sensors. The fitness function used in the algorithms is calculated based on coverage of all sensors in the region of interest (ROI). A simulating program for both surface types and all the three algorithms has been developed using MATLAB.

In all the three PSO, GA and HA evaluations, we found that the HA has exceeded PSO with a percentage of 26.12% and GA with 1.58% on rough surface. Similarly when we also compared the results of these algorithms on smooth surface, HA has exceeded PSO with a percentage of 22.24% and GA with 3.42%. Next to the HA, GA has a very good performance than PSO.

Key words: Particle Swarm Optimization, Genetic Algorithm, WSNs, 3-D terrain, Sensor Deployment.

1. Introduction

WSNs are tiny and low power devices which are used to gather and process environmental information using wireless communication. It has provided a bridge between the real physical and virtual worlds. WSNs allow the ability to observe the previously unobservable phenomena or events at a fine resolution over large spatio-temporal scales and have a wide range of potential applications to industry, science, transportation, civil infrastructure, and security.

Some sample applications are listed below:

- Seismic Monitoring
- Habitat and Ecosystem Monitoring
- Health Monitoring
- Monitoring Groundwater Contamination
- Rapid Emergency Response
- Industrial Process Monitoring
- Perimeter Security and Surveillance
- Automated Building Climate Control

In recent years there has been increasing interest in the field of WSNs. A WSNs consists of a number of wireless sensor nodes. These nodes are characterized by being very small in size with limited energy usually supplied by a battery. They communicate via built-in antennae over radio frequency signals. Sensor networks are composed of cooperating sensor nodes that can perceive the environment to monitor physical phenomena and events of interest.

Sensor networks can be used for target tracking, environmental monitoring, system control, and chemical or biological detection. In military applications, sensor networks can enable soldiers to see around corners and to detect chemical and biological weapons long before they get close enough to cause harm. Civilian uses include environmental monitoring, traffic control, and providing health care monitoring for the elderly while allowing them more freedom to move about [10]. One of the most active research fields in WSNs is that of coverage. Coverage is usually interpreted as how well a sensor network can monitor a field of interest. It can be thought of as a measure of quality of service. Coverage can be measured in different ways depending on the application. [10], A higher degree of coverage requires multiple sensors to monitor the same

location in order to produce more reliable results. Existing research focuses on coverage in the context of energy conservation. Some of them have proposed techniques to select the minimal set of sensor nodes to be active to maintain coverage [6].

The emerging WSNs have been envisioned for a diversity of autonomous monitoring and actuation applications on 2D plane, 3D volume and 3D surface. With its substantial impact on network performance, sensor deployment (SD) has been a central problem in wireless sensor network research [3]. In WSNs, sensors are randomly deployed in the sensor field which brings the coverage problem. One of the most critical issues of WSNs is the deployment in order to achieve maximum coverage on a terrain. It is a unique problem and in maximizing coverage, 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 (QoC). This can be achieved with minimum number of sensor nodes having maximum coverage in the network and the nodes are within the communication range [2].

1.1. Objective

1.1.1 General Objective

The general objective of this thesis work is to develop an optimal WSNs deployment solution of nodes on 3D surfaces to maximize area coverage under constrained number of nodes.

1.1.2 Specific objectives

- Investigate existing population based algorithms such as PSO, APSO and GA based on literature.
- Select the two better algorithms from the above listed.
- Integrate the two selected algorithms and form a HA.
- Simulate HA for WSNs node deployment using MATLAB developed program
- Analyze and evaluate the simulation result.

1.2 Statement of Problem

With the recent technological advances in wireless communication and networking, coupled with the availability of intelligent and low-cost actor and sensor devices with powerful sensing, computation, and communication capabilities, WSNs are about to enter the mainstream. Today,

one could easily envision wide range of real-world WSN-based applications from sensor-based environmental monitoring, home automation, health care, security, and safety class of applications, thereby promising to have a significant impact throughout our society. WSNs are comprised of a large number of sensor devices that can communicate with each other via wireless channels, with limited energy and computing capabilities.

There are different active research issues in WSNs. One of the fundamental issues that arise in WSN is coverage area in addition to location identification, tracking, and deployment. In this coverage, the nodes have the effective responsibility to cover the predefined area. The most effective approach of sensor deployment is to place sensors in such a manner that the maximal network coverage is achieved.

There are a number of research works that have dealt with WSN deployment using different techniques for 2D surfaces, for 3D volumes and recently for 3D geographical surfaces or terrains. Some of the research works has used population based artificially intelligent algorithms like genetic GA, PSO, and APSO etc.

Although the issue of WSN deployment and coverage problem is dealt with these deferent research works, most of these works were done on 2D plane and 3D volume coverage and very few worked on 3D surfaces while to the best of my knowledge none of them has also tried to implement the hybrid of two algorithms, on sensor deployment which might result in more optimal solution in WSNs deployment problem.

In our thesis work, the hybrid of two selected artificially intelligent algorithms has been used to form hybrid of the two algorithms to exploit the best features of the algorithms and to optimize the WSNs deployment further. Under constrained number of sensor nodes in a given model of terrain, we worked to maximize the area coverage of the sensors by using the HA.

1.3. Thesis Outline

This thesis is organized as follows. The first chapter gives the introduction of the overall thesis work followed by literature review in chapter two. Sensor deployment and design alternatives as well as design assumptions are presented in chapter three. Analysis and simulation results are discussed in chapter four. Finally, conclusion and future works are presented in chapter five.

2. Literature Review

According to the book given in [9], due to the nature of WSNs, there are new research challenges related to the design of algorithms and network protocols that will enable the development of sensor-based applications. According to most of the available literature in this emerging technology concentrates on physical and networking aspects of the subject. However, in most of the literature, a description of fundamental distributed algorithms that support sensor and actor devices in a wireless environment is either not included or briefly discussed [9]. The efficient and robust realization of such large, highly dynamic and complex networking environments is a challenging algorithmic and technological task [9].

In [10] states, deployment is a fundamental issue in WSNs that affects many facets of network operation, including routing, power management, security, coverage etc. According to it, broadly; there are two categories of deployments in WSNs:-these are random and deterministic deployments. Random deployments are typically the cases when the deployment area of the mission is physically inaccessible (e.g., volcanic, seismic zones etc.).Deterministic deployments on the other hand are more likely (and even preferable) in missions when the deployment area is physically accessible. The missions of deterministically deployed WSNs are increasingly becoming popular today [10].

As it is stated in the research work given in [8], the major challenge in designing WSNs is to find tradeoff between the desired and contrary requirements for the lifetime, coverage or cost while coping with the computation, energy and communication constraints. In [8] examines the optimal placement of nodes for a WSNs. It is impossible to consider the deployment of the nodes separately from WSNs applications. According to this research work [8], highlighted the properties of WSNs applications that determine the placement problem. It has identified and enumerated the various objectives that should be considered. In [8], provides an overview and concentrates on multi-objective strategies (such as, coverage, life time, energy efficiency, number of sensors etc.), their assumptions, and optimization problem formulation. The coverage problem is the objective that has been widely discussed in the literature. Assessing the coverage varies based on the underlying model of each sensor's field of view and the metric used to measure the collective coverage of

deployed sensors. The most commonly used sensor coverage model is a sensing disk model. All points within a disk centered at sensor are considered to be covered by the sensor. In [8], of WSNs, however, many papers assume a fixed sensing range and an isotropic detection capability of sensor. Results or a comparison between the various approaches for multi-objective nodes placement are presented.

The research work in [2] uses PSO to optimize sensor deployment for 2-D surface. Sensors are randomly deployed in the sensor field which brings the coverage problem [2]. According to it, sensor deployment is a unique problem to maximize 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. The main objective of this paper was to minimize the distance between the neighboring nodes, maximizing coverage in the network, while simultaneously satisfying all constraints [2].

PSO is a population based stochastic search technique introduced by Kennedy and Eberhart, inspired by social behavior of bird flocking or fish schooling [2]. It works in the same way as genetic algorithms and other evolutionary algorithms. Similar to evolution algorithm, PSO algorithm adopts a strategy based on particle swarm and parallel global random search. This algorithm determines search path according to the velocity and current position of particle without more complex evolution operation. PSO algorithm has better performance than early intelligent algorithms on calculation speed and memory occupation, and has less parameter and is easier to realize. All these algorithms update a set of solutions (called swarm in the context of PSO applying some operators and using the fitness information to guide the set of solutions for better regions of the search space. PSO differs from these algorithms by simulating the social behavior and moment dynamics of a swarm and by not employing a survival of the fittest model [2]. Each swarm always moves to the own local optimum solution and the global optimum solution. Finally, swarm finds the good optimum solution.

Common assumptions and parameters for many sensor network applications according to the research work in [2] are given below.

- All sensor nodes are homogenous and have mobility.

- They assumed that the deployed sensor nodes can fully cover the sensing field. Sensing coverage and communication coverage of each node is assumed to have a circular shape without any irregularity.
- The design variables are 2D coordinates of the sensor nodes.
- All the nodes cover equal sensing field areas.

Generally, it has used PSO to find the optimal positions of the sensors to determine the best coverage. 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. Finally results were presented which shows that, PSO has good effect in solving coverage problem [2].

The research work presented in [3] states that sensor deployment is a fundamental issue in a WSNs, which often dictates the overall network performance. Previous studies on sensor deployment mainly focused on sensor networks on 2D plane or in 3D volume. it has tackled the problem of optimal sensor deployment on 3D surfaces, aiming to achieve the highest overall sensing quality.

In [3], it is considered a given set of sensors deployed on general 3D surfaces with possible concave boundary condition. And it is assumed that the sensors can be deployed at any pre-determined locations. With the insights gained from conformal parameterization, it is proposed a series of practical algorithms to realize sensor deployment on general 3D surfaces that minimizes the network-wide unreliability of sensor data.

Before giving the formal definition of the optimal surface deployment problem studied in [3], it is first introduced the surface model and the wireless sensor network model, which are presented below, employed in the research [3].

Surface models adopted in this study are all topological disks. In other words, they are surfaces with single boundary. It can have complicated shapes and complex boundary conditions including concave case. They do not need to be single valued surfaces, and are not subject to any restriction on their height [3].

It is assumed that stationary and homogeneous sensors deployed on surfaces. And also it does not move after deployment and have the same sensing radius and identical sensing capacity. The

accuracy of their collected data depends on the distance between the sensor and the target point to be sensed [3].

In [1], WSNs are randomly deployed in the sensor field which brings the coverage problem. To maximize 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. It has used particle swarm algorithm and adaptive particle swarm optimization to cover the maximum volume possible for 3D with limited number of sensors. In this literature [1] there have been many previous efforts to provide optimal solution for deployment of sensors over a terrain and many evolutionary algorithms have been used to arrive at the solution. However, PSO has been successfully used in numerous engineering applications. APSO is modification of PSO which at each stage eliminates the invaluable or less valuable particles, which on evolution have undergone undesired process and have lost both local as well as global search capability [1]. In research [1] by taking inspiration from earlier researches and success of PSO in various fields, it has used APSO algorithm for its problem solution. Thus, with the help of this study it has proposed an efficient way of deployment of nodes in WSNs to cover the maximum volume possible for 3D terrains.

In this literature [1], it is assumed that all sensor nodes are homogeneous having same range, Sensors are stationery and Sensing coverage and communication coverage of each node are also assumed to have a spherical shape without any irregularity.

PSO is a relatively newer addition to a class of population based search technique for solving numerical optimization problems. The particles or members of the swarm fly through a multidimensional search space looking for a potential solution. Each particle adjusts its position in the search space from time to time according to the flying experience of its own and of its neighbors.

APSO in [1], is a modification of PSO which at each stage eliminates the invaluable or less valuable particles, which on evolution 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 simulation results illustrated the performance of APSO is better than PSO performance [1].

In [4], most of the studies in the literature today are proposed for 2D surfaces; however, real world sensor deployments often arise on 3D environments. The main objective of this paper was to maximize the overall QoC of a WSN when deploying a specific number of sensors on a 3D surface. Some GA approaches are empirically tested for the search of an optimal deployment scheme [4]. In this study,3D terrain types are used, *i.e.*, rough terrains, undulating terrains to smooth terrains. It has used a wavelet transform-based guided walk mutation algorithm for 3D terrains.

3. Sensor Deployment Algorithms and Design Assumptions

3.1. Introduction

Recently, WSNs have been studied intensively for applications such as monitoring physical environments. A WSNs is composed of many tiny, low-power nodes that integrate sensing units, transceivers, and actuators with limited on-board processing and wireless communication capabilities. These devices are deployed in a region of interest to gather information from the environment, which will be reported to a remote base station. WSNs have been considered in many potential applications for monitoring and collecting data, such as surveillance, biological detection, and traffic, pollution, habitat, and civil infrastructure monitoring. Sensor deployment (SD) is a critical issue since it reflects the cost and detection capability of a WSNs. [13].

SD is a fundamental issue to be solved in WSNs. A proper SD scheme can reduce the complexity of problems in WSNs as, for example, routing, data fusion, communication, etc. Furthermore, it can extend the lifetime of WSNs by minimizing energy consumption. A WSNs can be composed of homogeneous or heterogeneous sensors, which possess the same or different communication and computation capabilities, respectively. Although some works consider heterogeneous sensors, many existing works investigate node placement in the context of homogeneous WSNs. Less complexity and a better manageability are the most beneficial effects of homogeneity. Therefore, it is considered homogeneous nodes in WSNs in this thesis work [12]. These nodes can be deployed over a network in random or deterministic fashion. While the random SD is preferable in many applications for few number of nodes, if possible, other deployments should be investigated for larger number of nodes since an inappropriate SD can increase the complexity of other problems in WSNs. [12].

3.2. Sensor Deployment Algorithm

Today, one could easily envision a wide range of real-world WSN-based applications from sensor-based environmental monitoring, home automation, health care, security, and safety class of applications, there by promising to have a significant impact throughout our society. WSNs are comprised of a large number of sensor devices that can communicate with each other via wireless channels, with limited energy and computing capabilities. However, due to the nature of WSNs, they are witnessing new research challenges related to the design of algorithms and network protocols that will enable the development of sensor-based applications .In some applications, instead of throwing the sensor nodes on the environment (e.g., by airplane), they can be strategically placed in the sensor field according to a priori planning. Several challenges still need to be overcome to have ubiquitous deployment of sensor networks [10].

Sensors may be deployed in the field of region which brings the coverage problem. In order to overcome this problem, it has used algorithms. There are two broad categories of SD algorithm categories used, namely deterministic and opportunistic algorithms. Deterministic algorithms are used for less complex and fewer number of nodes deployment for less complex regions of interest. But for complex and high number of nodes with complex regions of interest, opportunistic deployment algorithms are more suitable. Among these opportunistic algorithms, population based intelligent algorithms are the most efficient algorithms.

Population-based optimization algorithms find near-optimal solutions to the difficult optimization problems by motivation from nature. A common feature of all population-based algorithms is that the population consisting of possible solutions to the problem is modified by applying some operators on the solutions depending on the information of their fitness. Hence, the population is moved towards better solution areas of the search space. Two important classes of population-based optimization algorithms are evolutionary algorithms and swarm intelligence-based algorithms. Although GA, Genetic Programming, Evolution Strategy and Evolutionary Programming are popular evolutionary algorithms, GA is the most widely used one in the literature [15].

In recent years, swarm intelligence has also attracted the interest of many research scientists of related fields. Bonabeau has defined the swarm intelligence as ‘‘. . . any attempt to design algorithms or distributed problem-solving devices inspired by the collective behavior of social insect colonies and other animal societies. . .”. It focused the viewpoint on social insects alone such as termites, bees, wasps and different ant species. However, a swarm can be considered as any collection of interacting agents or individuals. An ant colony can be thought of as a swarm whose individual agents are ants; a flock of birds is a swarm of birds. An immune system can be considered as a swarm of cells and molecules as well as a crowd is a swarm of people. PSO is also a population-based stochastic optimization technique and is well adapted to the optimization of nonlinear functions in multidimensional space. It models the social behavior of bird flocking or fish schooling. PSO has received significant interest from researchers studying in different research areas and has been successfully applied to several real-world problems [15].

3.2.1. PSO Algorithm

PSO is a nature inspired meta-heuristic method. It is inspired by the swarm behavior of birds flocking, and utilizes this behavior to guide the particles to search for globally optimal solutions [17].

Basically, in PSO, [17] a population of particles is spread randomly throughout the search space. The particles are assumed to be flying in the search space. The velocity and position of each particle is updated iteratively based on personal and social experiences. Each particle possesses a local memory in which the best so far achieved experience is stored. Also a global memory keeps the best solution found so far. The sizes of both memories are restricted to one. The local memory represents the personal experience of the particle and the global memory represents the social experience of the swarm. The balance between the effect of the personal and social experiences are maintained using randomized correction coefficients. The philosophy behind the velocity update procedure is to reduce the distance between the particle and the best personal and social known locations. PSO is very easy to implement and there have been many successful implementations in several real world applications. PSO is a population based heuristic approach. It can get stuck in local optima when dealing with complex multimodal functions. This is why

accelerating the convergence speed as well as avoiding the local optima are two primary goals in PSO research.

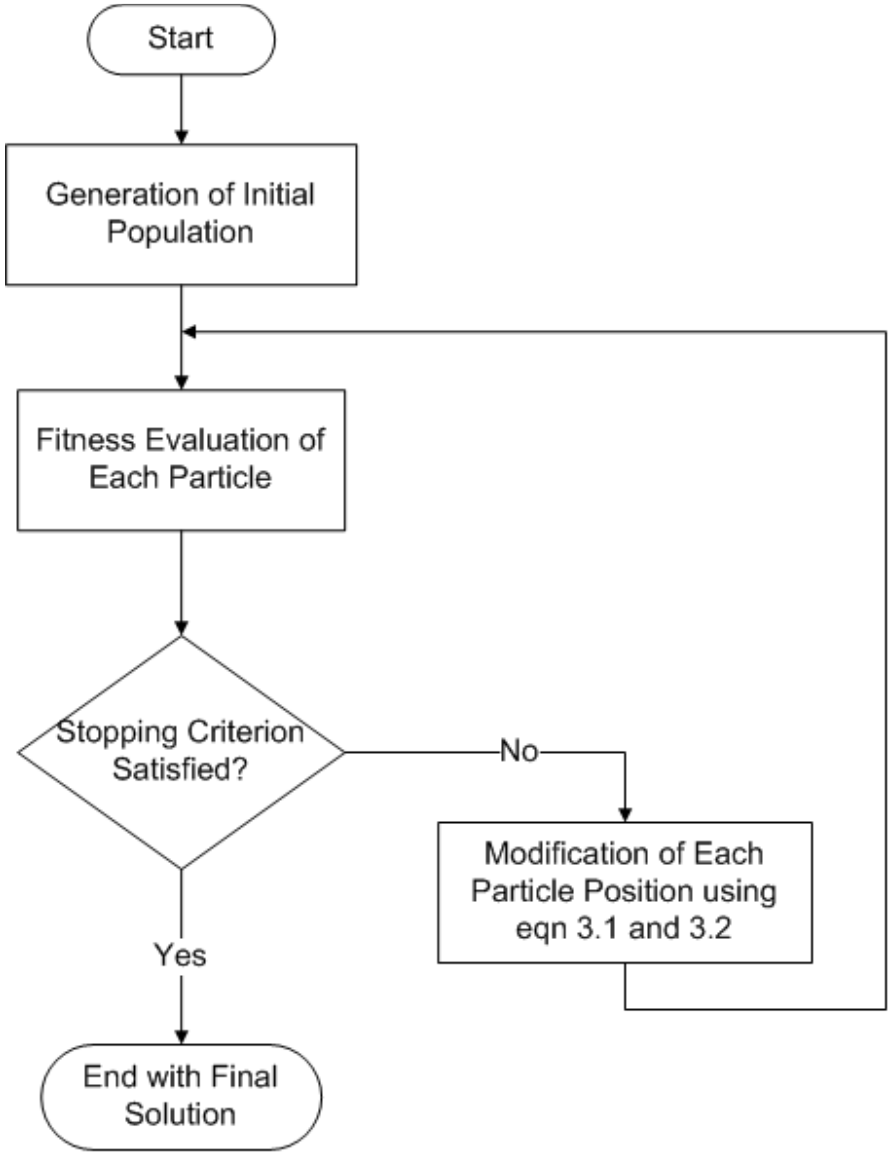

Figure 3.1 PSO Algorithm

In PSO, a population of particles starts to move in search space by following the current optimum particles and changing the positions in order to find out the optima. The position of a particle refers to a possible solution of the function to be optimized. Evaluating the function by the particle's position provides the fitness of that solution. In every iteration, each particle is updated by

following the best solution of current particle achieved so far ($p(t)$, particle best) and the best of the population ($g(t)$, global best). When a particle takes part of the population as its topological neighbors, the best value is a local best. The particles tend to move to good areas in the search space by the information spreading to the swarm. The particle is moved to a new position calculated by the velocity updated at each time step t. This new position is calculated as the sum of the previous position and the new velocity which is expressed in equation 3.1:

$$x(t+1) = x(t) + v(t+1) \quad \text{(3.1)}$$

Where $x(t)$ =previous position old position

$x(t+1)$ =New position

$v(t+1)$ =New velocity

$v(t)$ =Old velocity

$$v(t+1) = \omega v(t) + \Phi_1 rad(0,1)\big(p(t) - x(t)\big) + \Phi_2 rad(0,1)\big(g(t) - x(t)\big) \quad \text{(3.2)}$$

The parameter ω is called the inertia weight and controls the magnitude of the old velocity $v(t)$ in the calculation of the new velocity, whereas Φ_1 and Φ_2 determine the significance of $p(t)$ and $g(t)$, respectively. Furthermore, $v(t)$ at any time step of the algorithm is constrained by the parameter v_{max}. The swarm in PSO is initialized by assigning each particle to a uniformly and randomly chosen position in the search space. Velocities are initialized randomly in the range(v_{min} , v_{max}).

3.2.2. GA Algorithm

GA, in [19], is a computational model that solves optimization problems by imitating genetic processes and the theory of evolution. Solutions from a population are used to form a new population. This is motivated by the hope that the new population will be better than the old one. Solutions that will form new solutions are selected according to their fitness: the more suitable they are, the more chances they have to reproduce. This is repeated until some condition (for example, number of generations or improvement of the best solution) is satisfied. In the traditional GA, all the variables of interest must first be encoded as (binary) digits (genes) forming a string (chromosome).To minimize a function $f(x_1 , x_2 , \dots\dots x_k)$ using GA, first, each $x_i$is coded as a binary or floating-point string of length M Where:

x_1 = [10001…01001]

$x_2 = [00101…11110]$

…….

$x_k = [11110…01011]$

Where $[x_1 , x_2 , \ldots\ldots x_k]$ is called a chromosome and $x_i$are genes. Then three standard genetic operations, i.e., reproduction, crossover, and mutation are performed to produce a new generation. Such procedures are repeated until the pre-specified number of generations is achieved, or the required accuracy is satisfied.

Other coding types have been considered for the representation issue, such as Real Coded Genetic Algorithms, which would seem particularly natural when tackling optimization problems of parameters with variables in continuous or discontinuous domains. In the real-coded GAs, a chromosome is coded as a finite-length string of the real numbers corresponding to the design variables. The real-coded GAs is rigorous, precise, and efficient because the floating point representation is conceptually close to the real design space. In addition, the string length reduces to the number of design variables. A comparative study has concluded that the real-coded GAs outperformed binary-coded GAs in many optimization problems [19].

Almost all conventional optimization techniques search from a single point (centralized) but, GAs always operate on a whole population of points. It improves the chance of reaching the global optimum and also helps in avoiding local stationary point [20]. GA optimization technique is easy to discover global optimum but has slow convergence.

A Basic GA flow chart

- [START] Generate random population of n chromosomes (suitable solutions for the problem)
- [FITNESS] Evaluate the fitness f(x) of each chromosome x in the population
- [NEW POPULATION] Create a new population by repeating following steps until the new population is complete:
    - [Selection] Select two parent chromosomes from a population according to their fitness.
    - [Crossover] Cross over the parents to form new offspring (children).

- [Mutation] With a mutation probability, mutate new offspring at each locus (position in chromosome)
- [Accepting] Place new offspring in the new population.

- [REPLACE] Use new generated population for a further sum of the algorithm.
- [TEST] If the end condition is satisfied, stop, and return the best solution in current population.
- [LOOP] Go to step2 for fitness evaluation.

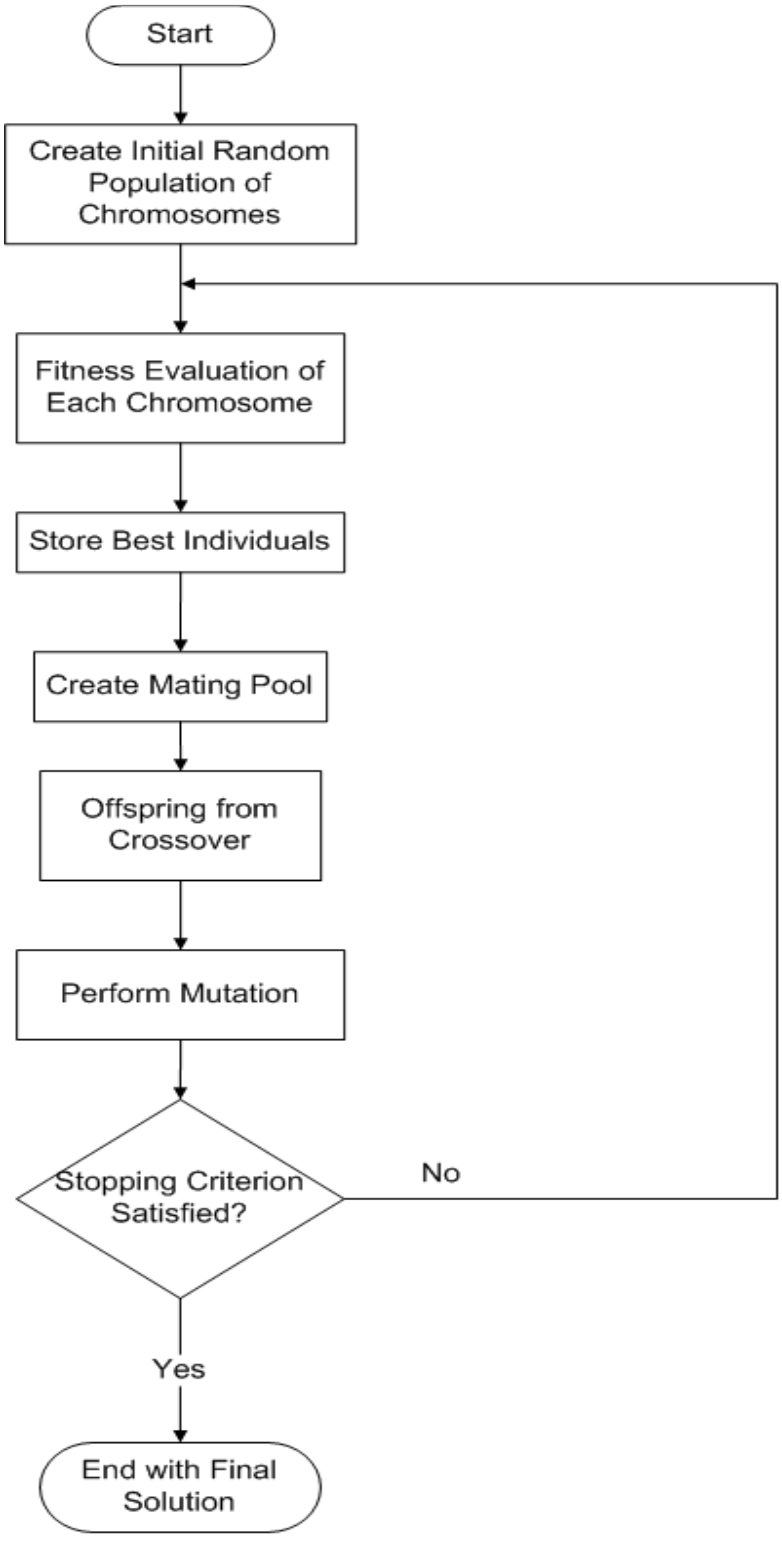

Figure 3.2. Genetic Algorithm

3.2.3. Hybrid of GA and PSO (HGAPSO) Algorithm

The hybrid of a GA with existing algorithms can always produce a better algorithm than either the GA or the existing algorithms alone [18]. In our thesis work, we proposed the Hybrid of GA and PSO (HGAPSO) algorithm, while to the best of my knowledge none of them has also tried to implement the hybrid of two algorithms, on WSNs deployment, to optimize or maximize the coverage area of all nodes under the constraint of sensor node on 3D surface. HGAPSO introduces the concept of the maturing phenomenon in nature into the evolution of individuals originally modeled by GA.

In GA the chromosomes may be trapped before becoming maturing. This problem can be avoided by embedding the selection mechanism of PSO in to GA.

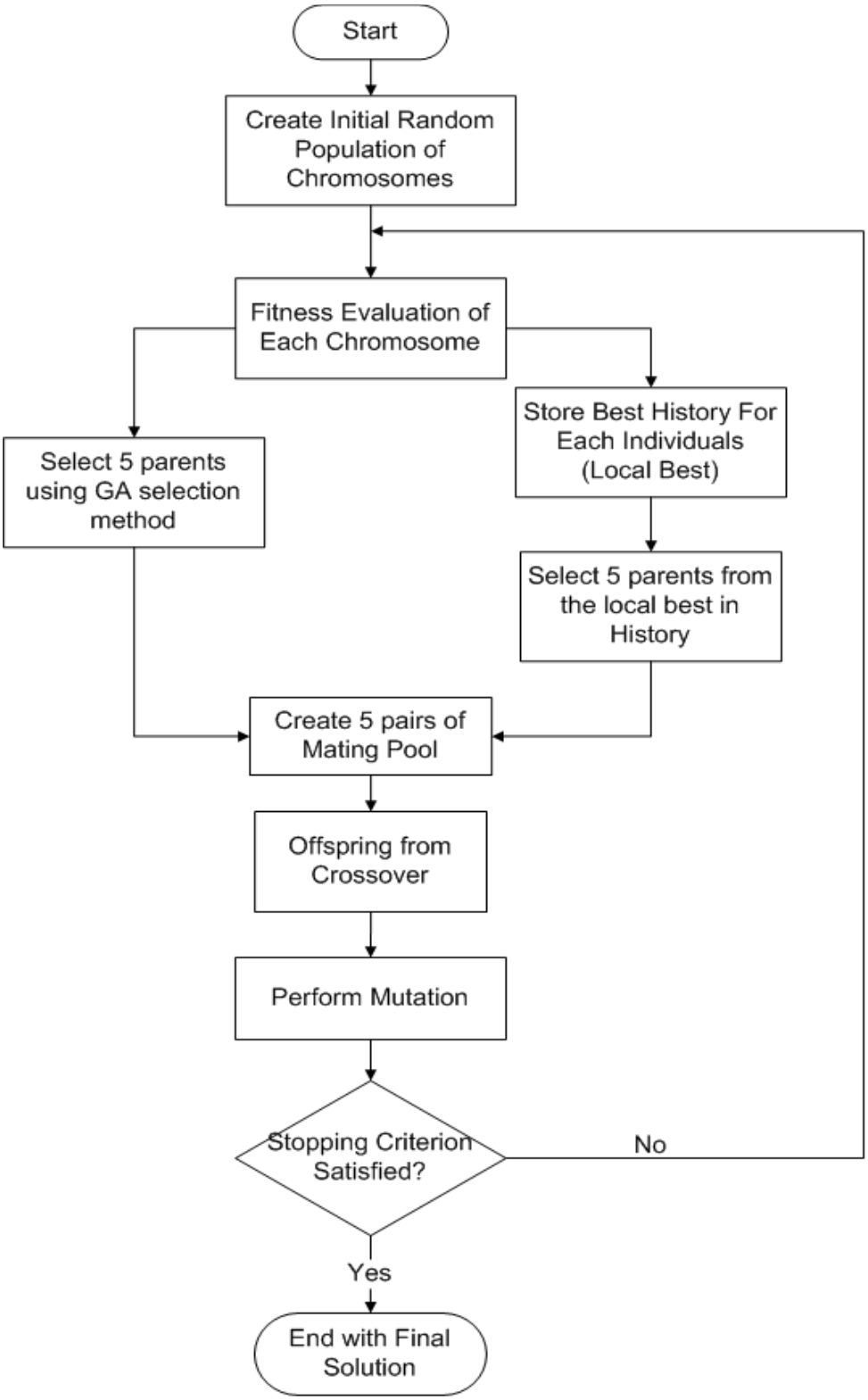

Figure 3.3. The flow chart of Hybrid Algorithm

3.3. Deployment Assumptions

When we are working on deployment of sensors, there are a number of design choices we should decide before we proceed to the main work. Some of these design choices we considered are given as follows:

1. The first design choice we considered is sensing range in order to calculate and optimize the area coverage of each of the sensors deployed in a region that needs to be monitored using the sensors. Sensors can have wide range of sensing range depending on the type of functions they are assigned. It ranges from few millimeters to hundreds of meters. We have taken sample sensing range and the sensing range of the nodes is considered to be 30m up to 50m. The range is just selected as being about the mean of the sensing ranges. Typically we have selected 41m sensing range just selected from the range for demonstration as we cannot show all possible ranges. Additionally, we have selected this range in such a way that the range can't give complete coverage of the area that need to be monitored.

2. The second point is the size of the region that needs to be monitored. Regardless of the topography of the region we select for all types of the surfaces, we have selected the size of the projection of the surface onto the x-y plane or horizontal to be in the range of between 300mx300m up to 500mx500m. Typically we have taken a region whose projection onto the horizontal plane is about 400mx400m.

3. The third point to be considered is the number of nodes over that area. We have done to optimize or maximize the sum of all coverage area of all nodes under the constraint of node scarcity. When there is less number of nodes than required to make complete coverage of the region monitored, there arises the optimal deployment of nodes so that the sensing coverage of each of the nodes gets maximum. So the number of nodes is selected in such a way that there is less number of nodes than required. About 35 to 50 sensors are used in this work. Typically we have selected 40 nodes for our simulation.

4. The fourth point we considered is the mobility of nodes. For simplicity we have considered all nodes to be static. If there is a need to move the nodes, prior programmed deployment and positioning shall be considered.

5. The fifth condition is the connectivity of the nodes to the destination. Each of the nodes should not be isolated so that they can transfer the sensed data to the sink or destination. Here the communication range is considered to be longer to reach the sink and it is assumed the sink is deployed in high altitude so that line of sight between the source node and sink is attained.

6. The sixth and last condition is the type of surface we use for simulating the deployment. We have considered function generated surface with close resemblance to the geographical surface. We used two surfaces i.e. one rough surface and the other relatively smooth. The surfaces are shown below in the following two figures.

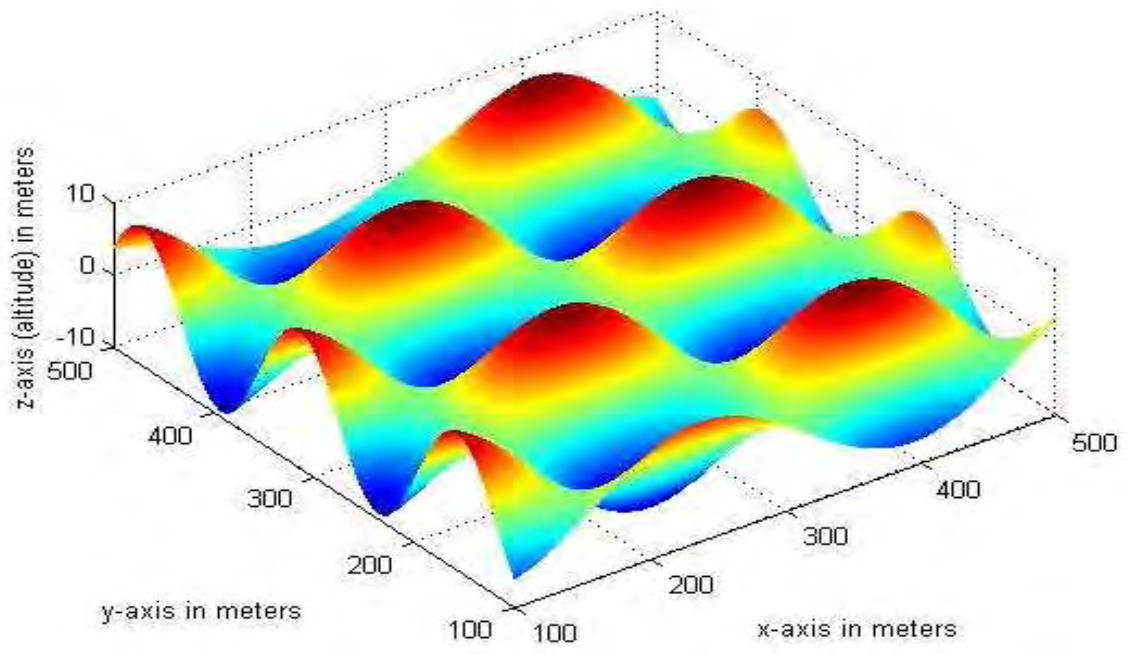

Figure 3.4. Examples of terrain types used: A rough terrain (z=10*sin (0.02x) cos (0.04y))

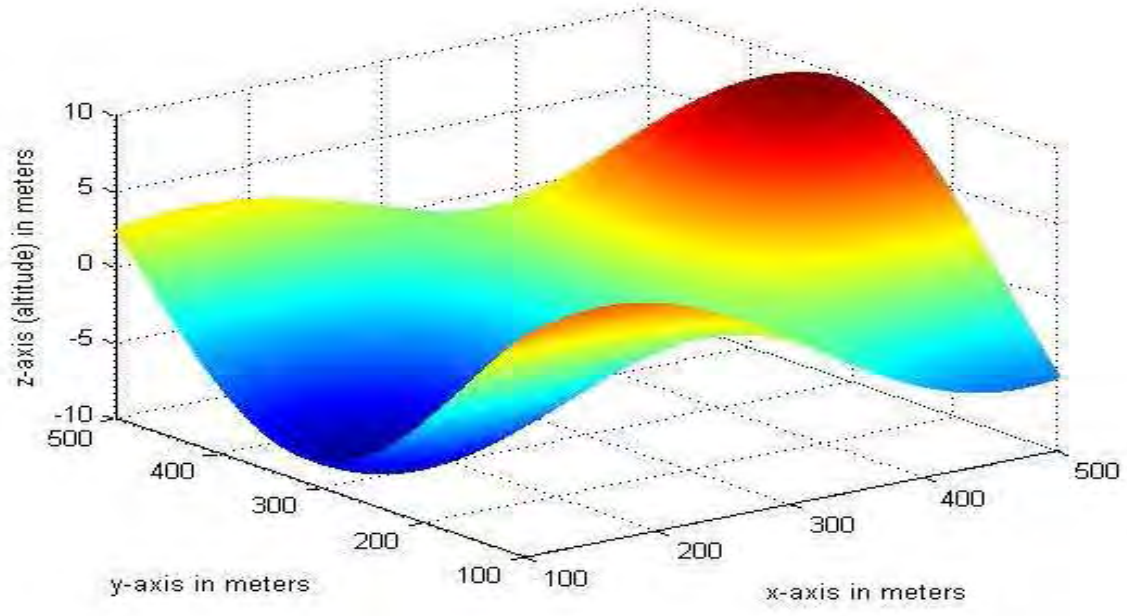

Figure 3.5. Examples of terrain types used: A smooth terrain (z=10*sin (0.01x) cos (0.01y))

3.3.1. Fitness Function

The fitness function used in the optimization of the deployment of the nodes is formulated so as to maximize the sum sensing coverage area of all the nodes used in the monitored region. The optimal positions of sensors are searched using two selected population based algorithms. These algorithms use the fitness function to search the optimal locations of the nodes in the 3D surface used in the simulating program.

If we let the number of nodes as N, the fitness function used by the search algorithms is given as:

$F = \sum_{i=1}^{N} A_i$…….…….. (3.3)

Where A_i is the area coverage of each of the nodes deployed in the monitored region.

The basic task in using the fitness function is to make calculations of coverage areas of each of the nodes. The conditions that we considered when the area calculations of each of the nodes are made. These conditions we considered are:

1. The line of sight (LOS) of each of the regions that are in the vicinity of the nodes should be carefully considered.
2. The overlapping sub-regions of the sensors have to be avoided or at least the overlapped area should not be considered more than once.

Using the above two principles we use grid based area calculation. The monitored area is divided into small grids (squares) which can be sensed or left unreached completely. We have considered the smallest unit of grid to be 4mx4m square grid. So the total area of the monitored region is subdivided to a 4mx4m small grids.

The program which calculates the coverage area of all the nodes after being deployed analyzes each of the grids if they are within the sensing range and in the LOS of one or more of the sensors being deployed. If the considered grid is within the range and in the LOS of at least one of the nodes, that grid will be marked as one (1). If none of the nodes reach that grid due to sensing range distance or being out of sight or none line of sight (NLOS), then the grid will be marked as zero (0). Finally the program counts the number of those grids marked one (1) and puts coverage area as number of grids being sensed. The typical grids in the monitored regions of the rough and smooth surfaces are given in the following figures.

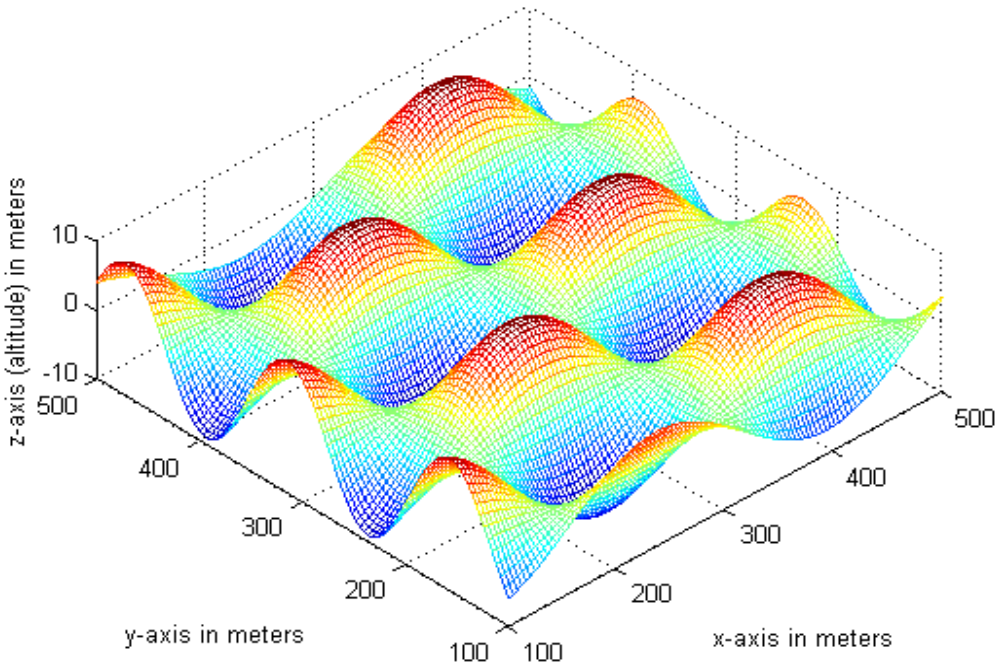

Figure 3.6. Grids of the rough surface

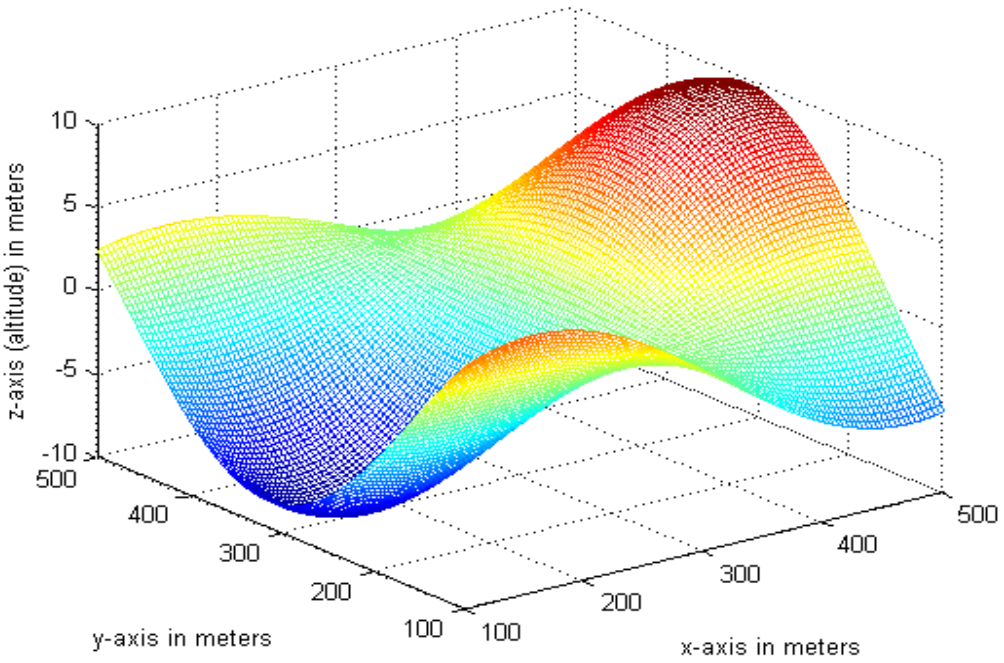

Figure 3.7. Grids of smooth surface

4. Simulation Results and Analysis

4.1. Introduction

In this chapter simulation results and analysis are discussed and presented. And also discussions on the coverage area performance of HA on two types of surfaces is presented. In Section 4.2, 4.3 and 4.4 simulation parameters and assumptions are stated. Coverage area of PSO, GA and HPSOGA on smooth and rough surface are also discussed in section 4.2, 4.3 and 4.4 respectively. Finally, performance comparison of PSO, GA and HA on smooth and rough surface are presented in section 4.5.

4.2. PSO Algorithm

As discussed in chapter 3, PSO is population based algorithm. We have used the PSO to find the optimal locations of sensors so as to maximize the coverage area of the monitored terrain. The MATLAB code of the PSO is given in appendix A at the end of this document which is customized from the reference document in [21]. The parameters of the PSO used in this research work is given as follows.

- Number of particles are taken 40
- Size of region= Projected region onto the horizontal is 400m x 400m
- Cognitive parameter $c1 = 1$
- Social parameter $c2 = 4 - c1$
- Number of iterations = 200
- Population size = 20
- Sensing Range of Nodes = 41m
- Mobility of nodes: - For simplicity we have considered all nodes to be static.
- Type of surfaces that we use are smooth and rough surfaces.

The initial positions (population) and initial velocities of the particles are randomly initialized. Then the next generation of particles (population) and corresponding velocities are updated as per the equations of 3.1 and 3.2 respectively. Then the deployment simulating program has been executed for smooth and rough surfaces separately. The number of total grids are 10,000 with size

of 4m x 4m. From these total number of grids, the fitness function calculates the number of sensed grids for a given positions of all nodes for which the positions of all the nodes are contained in one particle. So the fitness of each particle will be compared as per the result of the fitness function.

The results of the PSO algorithm for each typical surfaces is discussed in the following sections of 4.2.1 and 4.2.2.

4.2.1. PSO Result in a Typical Rough Surface

A typical surface used in this research is given in figure 3.3 in chapter 3. The PSO code for the rough surface is run and has given the result shown in the following graph. The graph shows the best (global best) solution coverage area, the local best and population average of each iteration.

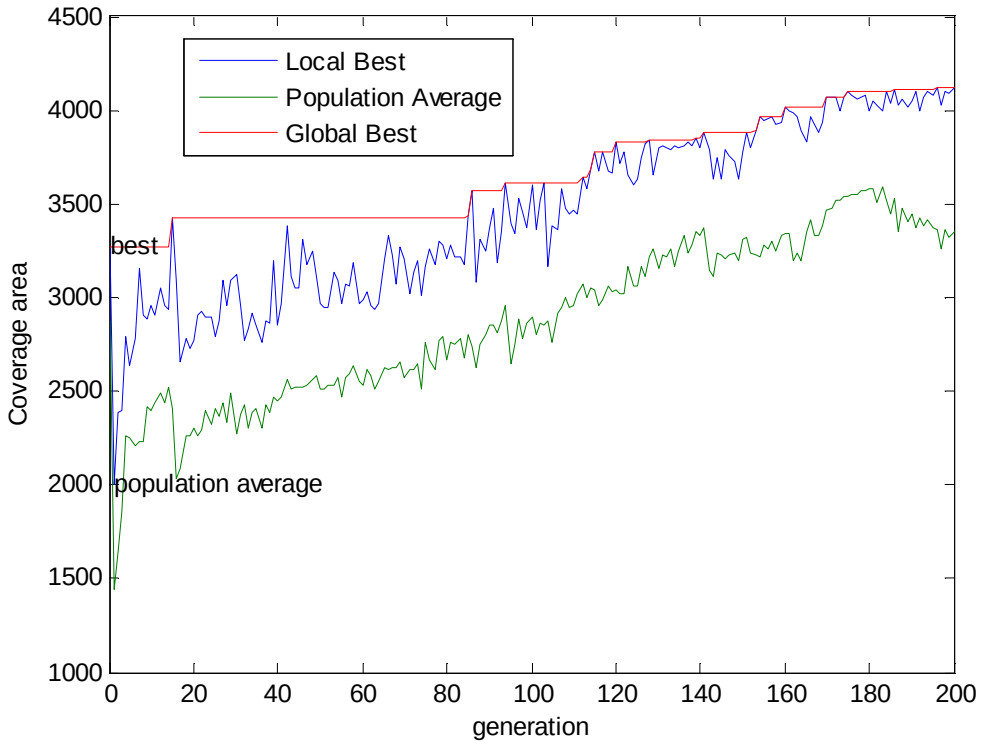

Figure 4.1 PSO Result for rough surface

The best and final area coverage out of 10,000 grids is 4,123 grids. The sensed grids plot is shown the figure below.

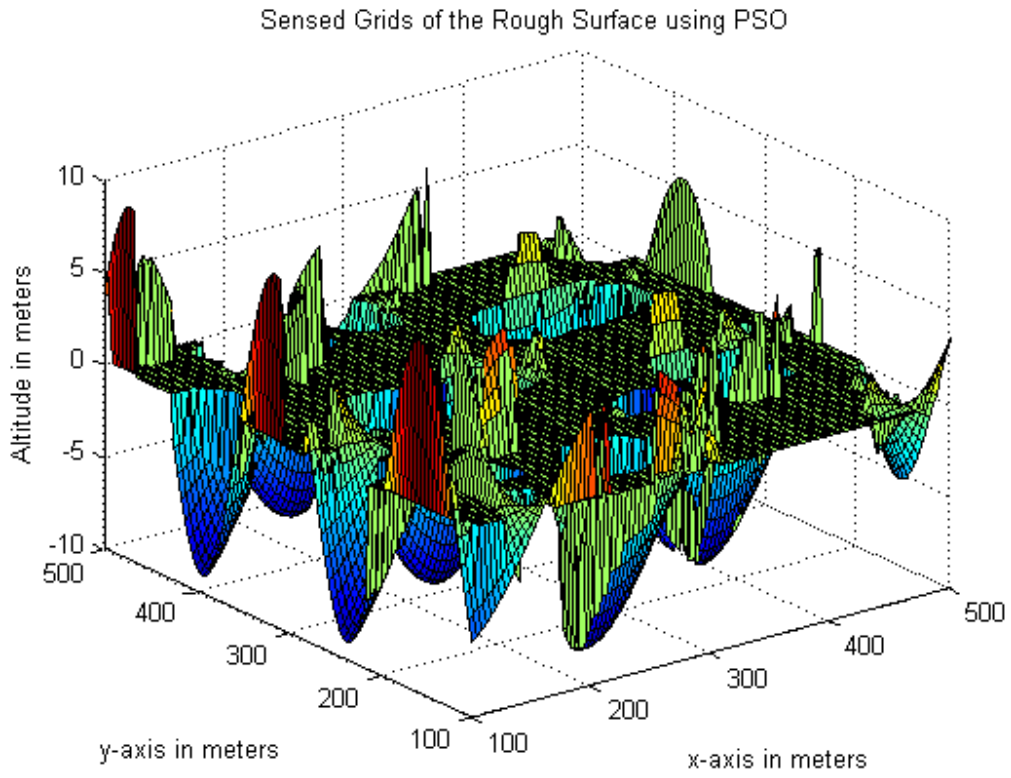

Figure 4.2 The plot of sensed grids of the rough surface. (The black is not sensed and plot with other colors is sensed)

The un-sensed grids are multiplied by zero and looks like simple black color plane on above figure. The final and the best solution gives rise to the number of grids being sensed to be 4,123 out of 10,000 grids which constitutes about 41.23% of the total coverage. The best positions for the sensor nodes of all 40s is given as per the result of the PSO algorithm.

Node No.	X-Coordinate	Y-Coordinate	Z-Coordinate
1	100	500	3.7107
2	100	165.2095	8.6164
3	100	393.8061	-9.0841
4	100	104.1174	-4.7349
5	101.0516	249.4142	-7.6672
6	101.7028	100	-5.8475
7	143.7057	148.224	2.4789
8	149.4564	250.5215	-1.2568
9	149.9766	499.5825	0.5993
10	164.5905	282.264	-0.4351
11	168.2898	100	1.4532
12	183.8701	424.9659	1.4115
13	187.5325	200.3169	0.9041
14	192.6968	475.1797	-6.4551
15	196.8935	315.1819	-7.1416
16	210.5335	499.4597	-3.75
17	212.38	100.4054	5.7325
18	213.1112	455.4606	-7.2693
19	242.4337	298.6202	-8.0543
20	261.4011	158.5367	-8.685
21	282.6749	111.6484	1.4367
22	293.9738	472.3994	-3.9241
23	294.3823	330.4566	-3.0629
24	303.7019	133.8509	-1.2427
25	331.0358	500	1.3514
26	341.3332	100	4.2299
27	344.978	209.3071	-2.8638
28	369.7083	393.5912	-8.9556
29	405.313	393.761	-9.6747
30	407.6026	408.7216	-7.6621
31	414.7356	261.1274	-4.7311
32	415.9249	211.8508	-5.1951
33	433.2212	396.3696	-6.8176
34	478.5206	107.6989	0.571
35	480.8435	268.2386	0.5019
36	493.3927	479.4169	-4.0599
37	499.9689	102.6747	3.093
38	500	162.7424	-5.3012
39	500	500	-2.2201
40	500	398.5625	5.2913

Table 4.1. Node Positions found using PSO for the typical rough surface

4.2.2. PSO Result in a Typical Smooth Surface

A typical surface used in this research is given in figure 3.4 in chapter 3. The PSO code for the smooth surface is run and has given the result shown in the following graph. The graph shows the best (global best) solution coverage area, the local best and population average of each iteration.

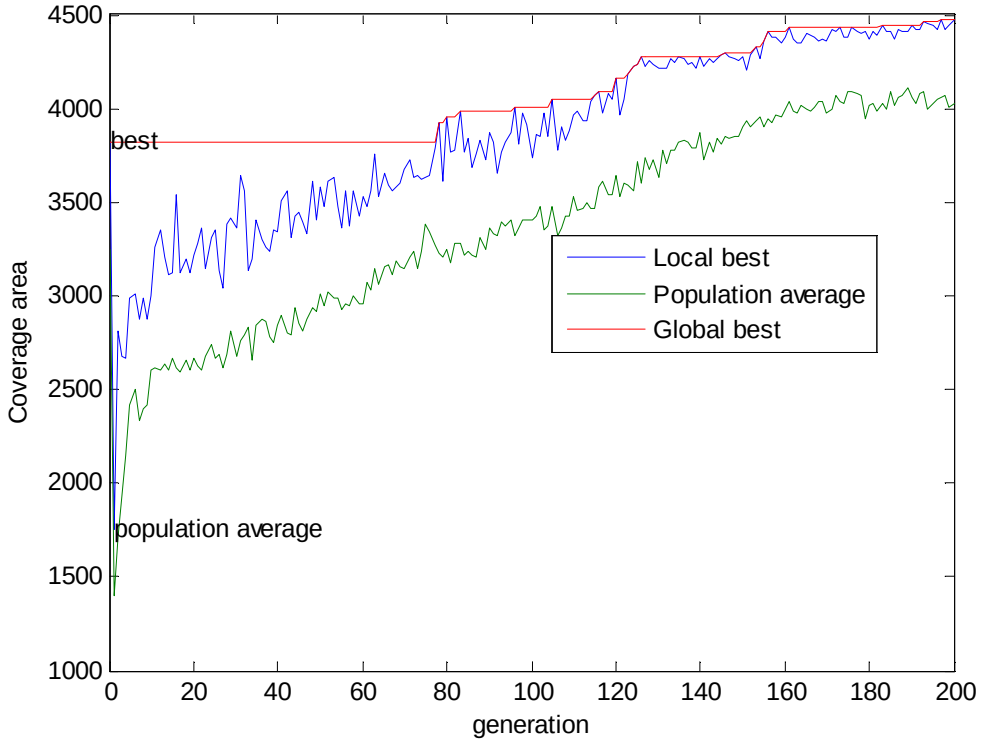

Figure 4.3 PSO Result for Smooth Surface

The best and final area coverage out of 10,000 grids is 4,473 grids. About 44.73% of the total area to be monitored is sensed as per the best solution of the PSO. The result for smooth surface is better than the rough surface due to improved LOS between node locations and the sensed grid. The smooth surface has improved about 8.49% when compared to the rough surface area coverage. This improvement is accounted totally for LOS improvement since the sensing range for both rough and smooth surface is the same. The sensed grids plot is shown the figure below.

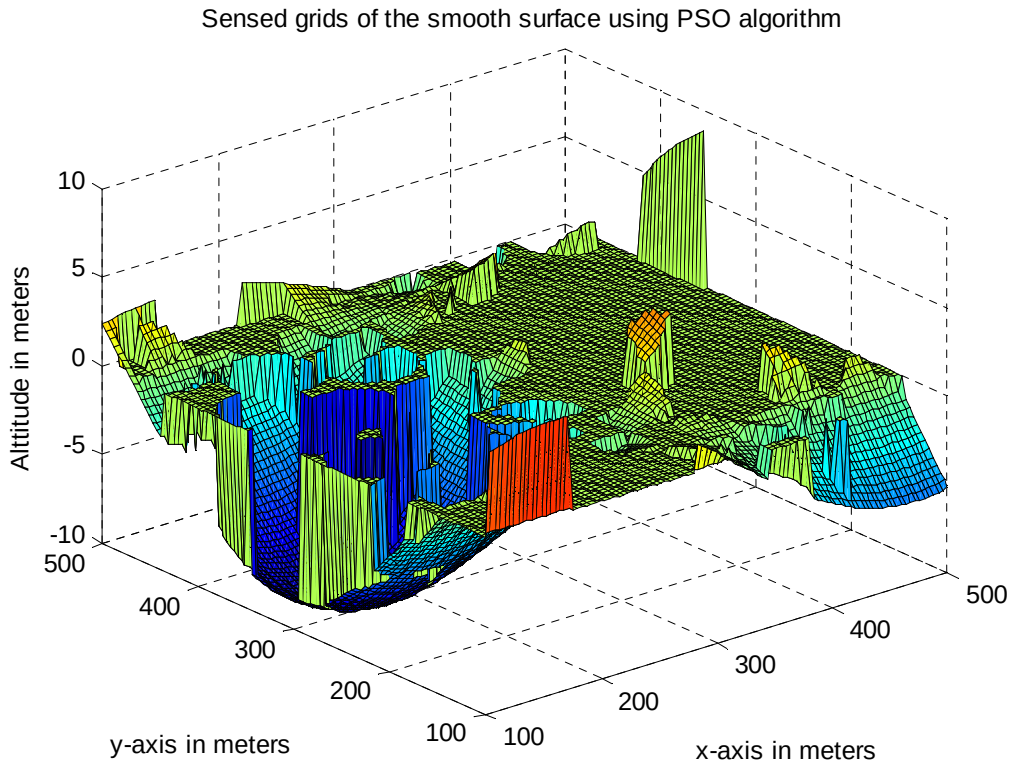

Figure 4.4 The plot of sensed grids of the smooth surface. (The black is not sensed and plot with other colors is sensed)

The un-sensed grids are multiplied by zero and looks like simple black color plane on above figure. The best positions for the sensor nodes of all 40s to be deployed on the smooth surface is given as per the result of the PSO algorithm as shown in the following table 4.2.

Node No.	X-Coordinate	Y-Coordinate	Z-Coordinate
1	100	500	2.3869
2	110.0306	500	2.5284
3	117.6515	187.0334	-2.7244
4	119.405	466.1555	-0.4725
5	125.1644	319.9521	-9.4791
6	135.331	254.6819	-8.0876
7	135.3631	367.8168	-8.3928
8	137.7189	100	5.3021
9	147.1925	465.3424	-0.5864
10	147.8579	182.8622	-2.539
11	176.2516	416.0642	-5.1457
12	179.7393	332.7964	-9.5756
13	194.8729	214.425	-5.0425
14	196.6107	269.3097	-8.316
15	239.0411	393.4099	-4.7916
16	241.9845	312.0312	-6.6055
17	244.2339	437.3114	-2.1421
18	252.7324	500	1.6349
19	263.3569	237.3499	-3.4987
20	288.7711	448.3672	-0.5695
21	292.9658	343.5598	-2.0133
22	295.4433	184.3123	-0.5005
23	309.6942	488.9253	0.0785
24	318.3826	103.9857	-0.2138
25	319.6289	453.371	0.0972
26	324.574	142.3823	-0.1522
27	342.1165	225.3947	1.7419
28	344.2324	112.4144	-1.2795
29	363.6655	464.406	0.3244
30	402.085	500	-2.185
31	406.6802	155.4257	-0.1321
32	412.6555	119.3619	-3.0688
33	454.6692	100	-5.329
34	455.8983	100	-5.3396
35	469.2367	133.3091	-2.3543
36	481.5761	389.511	7.2539
37	485.005	156.9438	-0.0135
38	488.4533	488.0978	-1.6531
39	495.8239	100	-5.2406
40	500	387.8654	7.1003

Table 4.2. Positions of Nodes for Smooth Surface found by PSO

4.3. GA Algorithm

As discussed in chapter 3, GA is also population based algorithm. We have used the GA to find the optimal locations of sensors so as to maximize the coverage area of the monitored terrain. The MATLAB code of the GA is given in appendix B at the end of this document which is customized from the reference document in [21]. The parameters and assumptions of the GA used in this research work is given as follows.

- Number of Particles are Taken = 40
- Size of Region= Projected region onto the horizontal is 400m x 400m
- Mating Factor (Selection Factor)= 0.5
- Mutation Rate = 0.02 (2%)
- Sensing Range of Nodes = 41m
- Number of Iterations = 200
- Selection Type = Steady State Selection
- Elitism = One best chromosome is copied to the next generation before any genetic operator gets operational (before any operation takes place too cerate the next generation)
- Crossover Type = Single Point Crossover
- Population size = 20
- Mobility of nodes: - For simplicity we have considered all nodes to be static.
- Type of surfaces that we use are smooth and rough surfaces.

The initial positions (population) of the chromosomes are randomly initialized. Then the next generation of particles (population) are updated by the genetic operators (crossover and mutation). Then the deployment simulating program has been executed for smooth and rough surfaces separately. The number of total grids are 10,000 with size of 4m x 4m. From these total number of grids, the fitness function calculates the number of sensed grids for a given positions of all nodes for which the positions of all the nodes are contained in one chromosome. So the fitness of each chromosome will be compared as per the result of the fitness function.

The results of the GA algorithm for each typical surfaces is discussed in the following sections of 4.3.1 and 4.3.2.

4.3.1. GA Result in a Typical Rough Surface

A typical surface used in this research is given in figure 3.3 in chapter 3. The GA code for the rough surface is run and has given the result shown in the following graph. The graph shows the best solution coverage area and population average of each iteration.

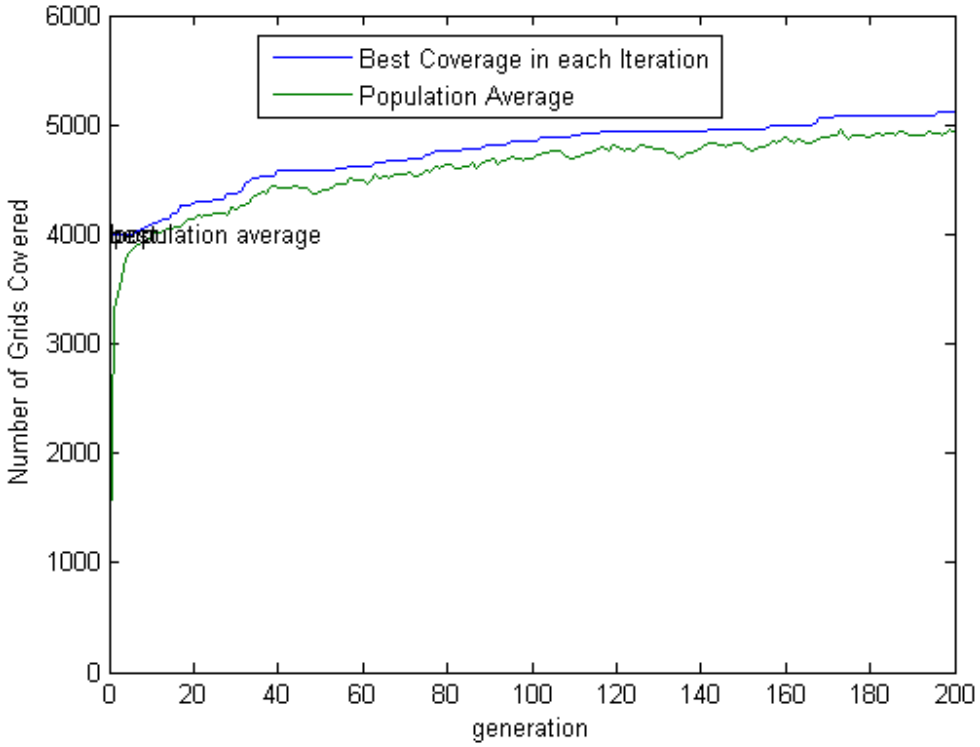

Figure 4.5 GA Result for Rough Surface

The best and final area coverage out of 10,000 grids is 5,119 grids. The sensed grids plot is shown the figure below.

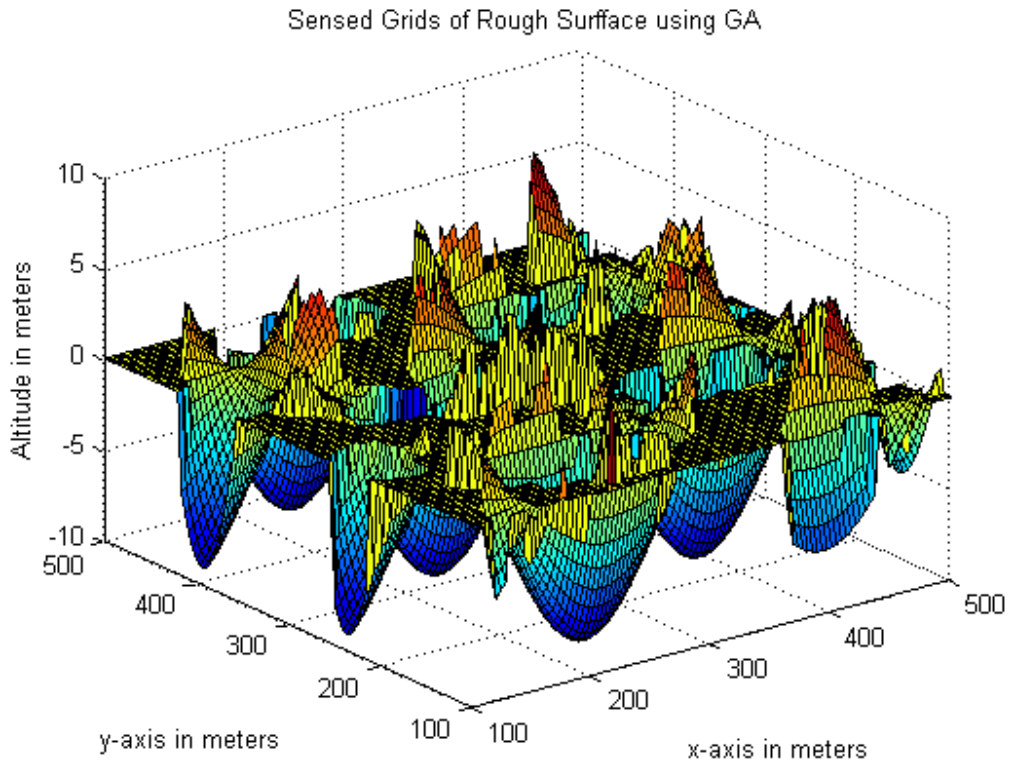

Figure 4.6 The plot of sensed grids of the rough surface. (The black is not sensed and plot with other colors is sensed)

The un-sensed grids are multiplied by zero and looks like simple black color plane on above figure. The final and the best solution gives rise to the number of grids being sensed to be 5,119 out of 10,000 grids which constitutes about 51.19% of the total coverage. The best positions for the sensor nodes of all 40s is given as per the result of the GA algorithm.

Node No.	X-Coordinate	Y-Coordinate	Z-Coordinate
1	118.9536	387.7314	-6.7715
2	130.1391	233.4849	-5.1125
3	147.6137	418.6123	-0.9583
4	152.1168	101.1989	-0.611
5	160.3103	360.2083	0.1729
6	164.1139	411.8133	1.012
7	168.4857	446.8291	-1.2665
8	169.132	177.5237	-1.6325
9	183.2082	325.0479	-4.5251
10	200.3382	173.5896	-6.0116
11	212.1216	471.2645	-8.9159
12	216.2241	127.3449	-3.4457
13	221.6831	293.6998	-6.5703
14	244.1603	181.6737	-5.4583
15	267.2399	141.9742	-6.6379
16	268.652	473.5753	-7.8615
17	269.1812	457.9522	-6.7504
18	271.6679	284.5825	-2.84
19	273.173	337.0291	-4.4585
20	303.0633	178.5393	-1.4387
21	307.2636	275.6628	-0.0425
22	309.014	356.4429	0.1235
23	319.7927	215.0099	-0.7634
24	330.2582	405.3472	-2.768
25	335.5114	434.0564	0.3454
26	362.564	411.26	-6.07
27	368.5399	232.8004	-8.799
28	389.0008	372.0698	-6.7658
29	401.1709	100.5248	-6.2848
30	418.2232	225.5945	-8.0329
31	420.622	257.3428	-5.4755
32	429.674	398.0634	-7.2186
33	447.2868	436.8245	0.8896
34	447.8575	114.574	-0.5818
35	459.6479	373.7428	-1.6681
36	460.4123	269.754	-0.4382
37	467.0626	192.16	0.1392
38	483.7546	342.7628	-1.0251
39	493.4664	458.9273	-3.7895
40	497.5167	145.2949	-4.4698

Table 4.3. Node Positions found using GA for the typical rough surface

The GA outperformed and very much exceeded the PSO when we compare the results of the two algorithms for the same rough typical surface used. We have also used the same iteration number and all other parameters that can be shared by the two algorithms. GA has exceeded PSO by about 996 grids with a percentage growth of about 24.15% as compared to PSO.

4.3.2. GA Result in a Typical Smooth Surface

A typical surface used in this research is given in figure 3.4 in chapter 3. The GA code for the smooth surface is run and has given the result shown in the following graph. The graph shows the best solution coverage area and population average of each iteration.

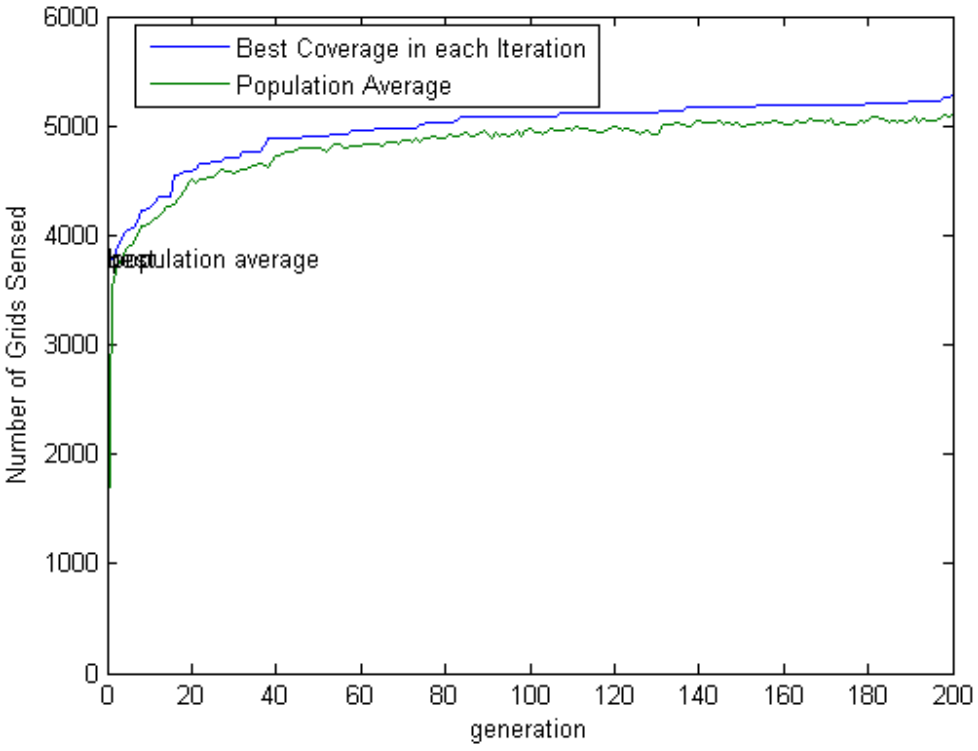

Figure 4.7 GA Result for Smooth Surface

The best and final area coverage out of 10,000 grids is 5,287 grids. About 52.87% of the total area to be monitored is sensed as per the best solution of the GA. The result for smooth surface is better than the rough surface due to improved line of sight between node locations and the sensed grid. The smooth surface has improved about 3.28% when compared to the rough surface area coverage. This improvement is accounted totally for LOS improvement since the sensing range for both

rough and smooth surface is the same as it is discussed in PSO results of rough and smooth surfaces under section 4.2.1 and 4.2.2. But the improvement between rough and smooth surfaces in PSO is better (more than 8%) than the improvement in GA (more than 3%). The sensed grids plot for smooth surface using GA is shown the figure below.

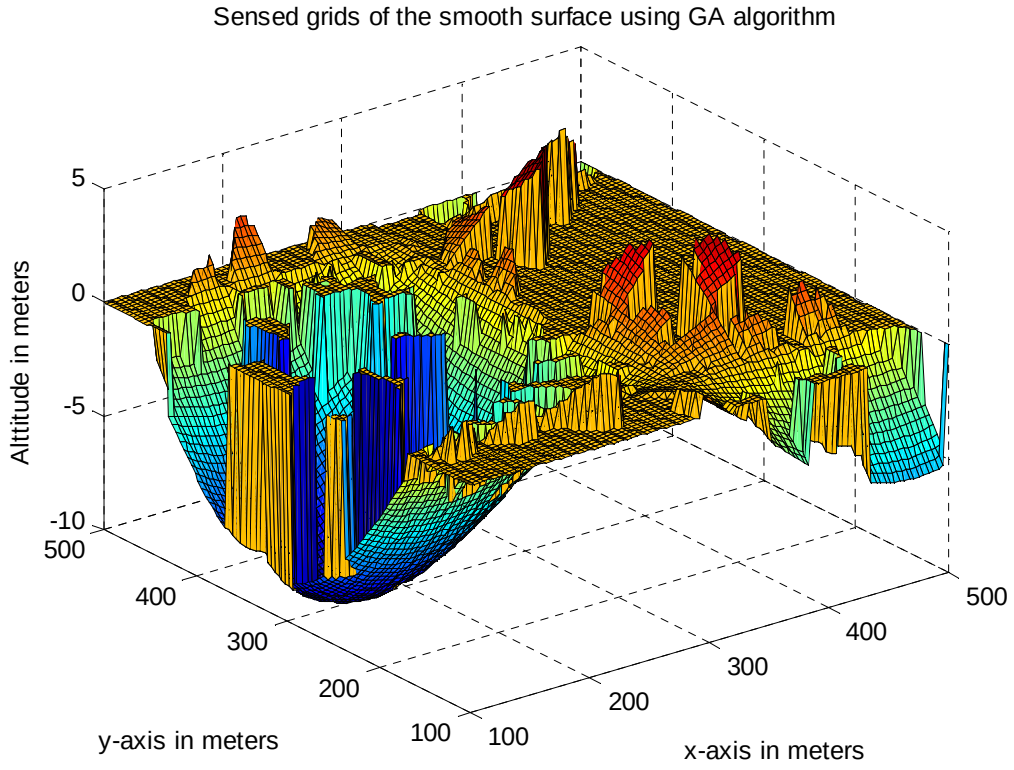

Figure 4.8 The plot of sensed grids of the smooth surface. (The black is not sensed and plot with other colors is sensed)

The un-sensed grids are multiplied by zero and looks like simple black color plane on above figure. The best positions for the sensor nodes of all 40s to be deployed on the smooth surface is given as per the result of the GA algorithm as shown in the following table 4.4.

Node No.	X-Coordinate	Y-Coordinate	Z-Coordinate
1	112.7808	169.704	-1.1375
2	117.5063	418.7664	-4.6226
3	120.026	392.7654	-6.5868
4	120.8452	208.426	-4.593
5	129.8487	275.6	-8.9243
6	147.9509	223.5846	-6.1453
7	155.3969	338.6049	-9.7013
8	176.4003	449.2347	-2.1421
9	184.1362	171.0711	-1.3438
10	193.2347	238.307	-6.7893
11	196.6332	365.3139	-8.0466
12	204.5519	298.9253	-8.7912
13	210.1463	407.0046	-5.1668
14	221.688	174.7691	-1.4051
15	221.7218	463.9728	-0.5795
16	243.8909	154.6132	0.1594
17	245.9955	284.4592	-6.0248
18	258.0592	354.3365	-4.8967
19	261.4433	246.6483	-3.9272
20	282.4896	431.1912	-1.2141
21	283.1531	312.4068	-3.0507
22	293.0723	467.7294	-0.0734
23	296.5336	400.6221	-1.1379
24	296.8154	232.9401	-1.1871
25	300.6707	367.7479	-1.1563
26	316.1134	199.0476	0.0796
27	316.1323	170.792	0.027
28	322.1638	121.3565	-0.2796
29	331.7776	235.3611	1.2362
30	342.4034	145.0085	-0.3356
31	354.2088	404.0118	2.428
32	361.0908	495.7375	-1.097
33	372.4257	191.2619	1.8445
34	380.0982	148.7204	-0.5115
35	382.3047	117.4128	-2.4337
36	410.4477	453.5906	1.4411
37	425.8567	466.8087	0.398
38	427.9032	147.8317	-0.8381
39	463.5905	495.3447	-2.3803
40	466.3743	117.2694	-3.8721

Table 4.4 Positions of Nodes for Smooth Surface found by GA

4.4. Hybrid Algorithm

The simulation parameters and assumptions are listed as follows:-

- Number of particles are taken 40
- Size of region=400*400 square meter projected onto the horizontal
- Mutation rate =0.02
- Sensing Range of Nodes = 41m
- Selection=0.5
- Mobility of nodes: - For simplicity we have considered all nodes to be static.
- Type of surfaces that we use are smooth and rough surfaces.

The initial positions (population) of the chromosomes are initialized by randomly. Then the next generation of chromosomes (population) are updated by the genetic operators (crossover and mutation). After the offspring are generated, the selection of the next generation uses the method of the PSO algorithm to compare and decide to update or transfer the old chromosome to the next generation. The comparison of the new offspring population with the older generation is based on the fitness function. The modification of the genetic algorithm using the selection method used by the PSO makes the algorithm hybrid. After making the comparison of the old generation and the new generation, the selection of the best from the new and old corresponding chromosomes will be made partially. All the next generation selection is not depending on the fitness function because of the high probable of being trapped in local minima. In this research, it has been experimented that local minima is the most probable condition to occur if total selection of the next members of the population is based on fitness function. The concept of local minima is introduced in this hybrid algorithm to hold on the best local chromosomes of the population for each corresponding chromosomes in the population. The comparison of the new GA operators generated offspring is made with this local minima.

Then the deployment simulating program has been executed for smooth and rough surfaces separately. The number of total grids are 10,000 with size of 4m x 4m. From these total number of grids, the fitness function calculates the number of sensed grids for a given positions of all nodes for which the positions of all the nodes are contained in one chromosome. So the fitness of each chromosome will be compared as per the result of the fitness function.

The results of the hybrid algorithm for each typical surfaces is discussed in the following sections of 4.4.1 and 4.4.2.

4.4.1. Hybrid Algorithm Result in a Typical Rough Surface

A typical surface used in this research is given in figure 3.1 in chapter 3. The hybrid algorithm simulating code for the rough surface is run and has given the result shown in the following graph. The graph shows the best solution coverage area and population average of each iteration.

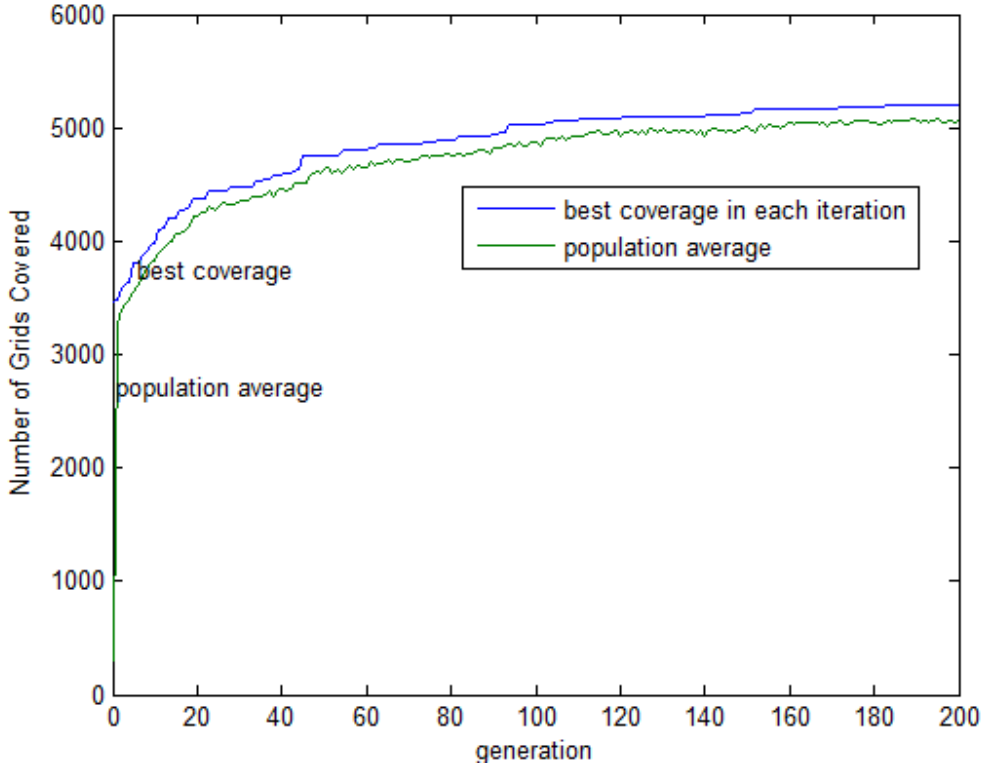

Figure 4.9 Hybrid Algorithm Result for Rough Surface

The best and final area coverage out of 10,000 grids is 5,200 grids. The sensed grids plot is shown the figure below.

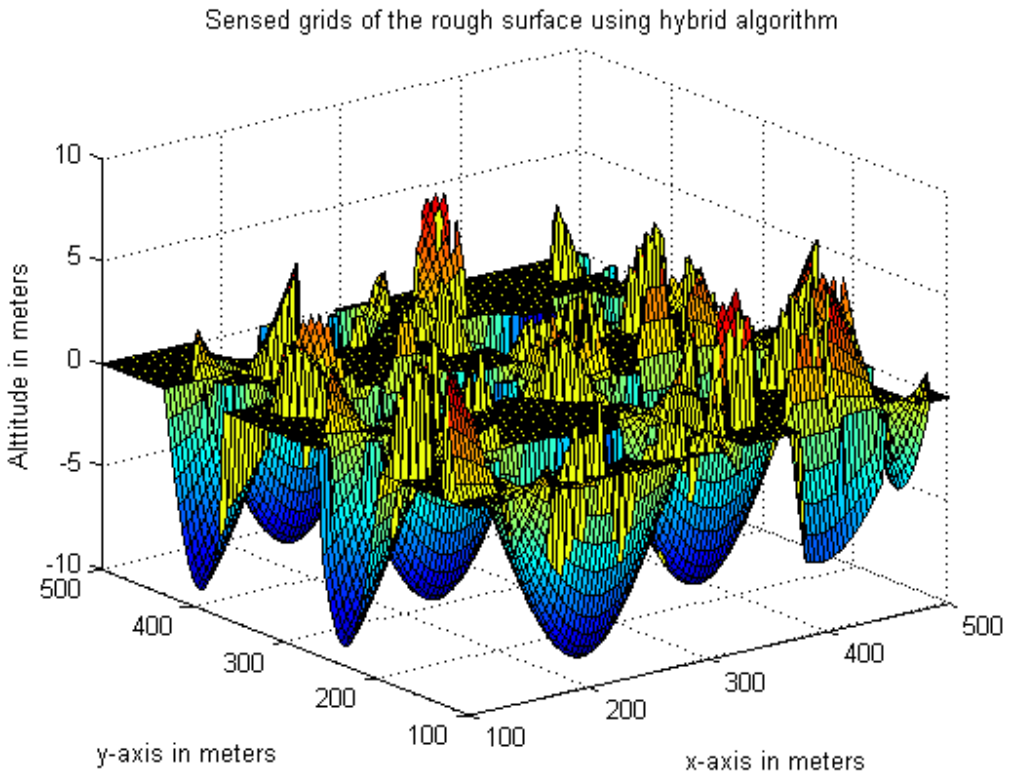

Figure 4.10 The plot of sensed grids of the rough surface. (The black is not sensed and plot with other colors is sensed)

The un-sensed grids are multiplied by zero and looks like simple black color plane on above figure. The final and the best solution gives rise to the number of grids being sensed to be 5,200 out of 10,000 grids which constitutes about 52.00% of the total coverage. The best positions for the sensor nodes of all 40s is given as per the result of the hybrid algorithm.

Node No.	X-Coordinate	Y-Coordinate	Z-Coordinate
1	116.4828	230.0136	-7.0746
2	119.0346	399.4889	-6.643
3	134.7867	198.7213	-0.4085
4	136.77	116.9884	-0.1298
5	146.6864	261.5467	-1.0499
6	159.7599	350.0733	-0.0717
7	164.6327	426.031	0.3541
8	168.03	280.8067	-0.5095
9	185.2295	327.4481	-4.6008
10	186.8397	121.8355	-0.899
11	187.0713	168.9445	-5.0211
12	208.3207	474.23	-8.4851
13	225.7235	156.3787	-9.8009
14	230.8468	289.6409	-5.5388
15	246.6142	327.7588	-8.3505
16	260.3715	460.9831	-8.0694
17	282.8591	141.6105	-4.7728
18	290.9826	493.0405	-2.876
19	299.5787	300.1306	-2.4341
20	318.9386	272.2466	-0.1007
21	322.8885	437.9126	0.4091
22	323.9908	173.2308	1.5599
23	326.4063	127.6597	0.9309
24	327.354	373.7128	-1.8917
25	343.7641	235.9263	-5.5806
26	350.8195	397.0119	-6.5932
27	384.9864	376.8818	-7.9686
28	402.8587	213.048	-6.0663
29	404.7739	258.1548	-6.025
30	412.0602	106.3881	-4.0848
31	442.1035	398.3419	-5.3633
32	445.2152	208.873	-2.3882
33	447.1455	258.4479	-2.8328
34	450.7791	359.1665	-0.9051
35	452.0918	115.5479	-0.3376
36	474.2634	200.6227	0.1028
37	476.0256	260.1258	0.5322
38	481.9406	151.091	-2.0634
39	484.3865	328.3748	-2.1903
40	485.5162	457.9446	-2.4278

Table 4.5 Node Positions found using HPSOGA for the typical rough surface

The HA outperformed and very much exceeded the PSO when we compare the results of the two algorithms for the same rough typical surface used and other parameters being set the same. We have also used the same iteration number and all other parameters that can be shared by the two algorithms. HA has exceeded PSO by about 1077 grids with a percentage growth of about 26.12%. When we also compare the results of Hybrid and GA algorithms for the same rough typical surface used, with the same iteration number and all other parameters that can be shared by the two algorithms, HA has exceeded GA by about 81 grids with a percentage growth of about 1.58%. This improvement is a significant one brought about by the HA.

4.4.2. Hybrid Algorithm Result in a Typical Smooth Surface

A typical surface used in this research is given in figure 3.4 in chapter 3. The Hybrid algorithm code for the smooth surface is run and has given the result shown in the following graph. The graph shows the best solution coverage area and population average of each iteration.

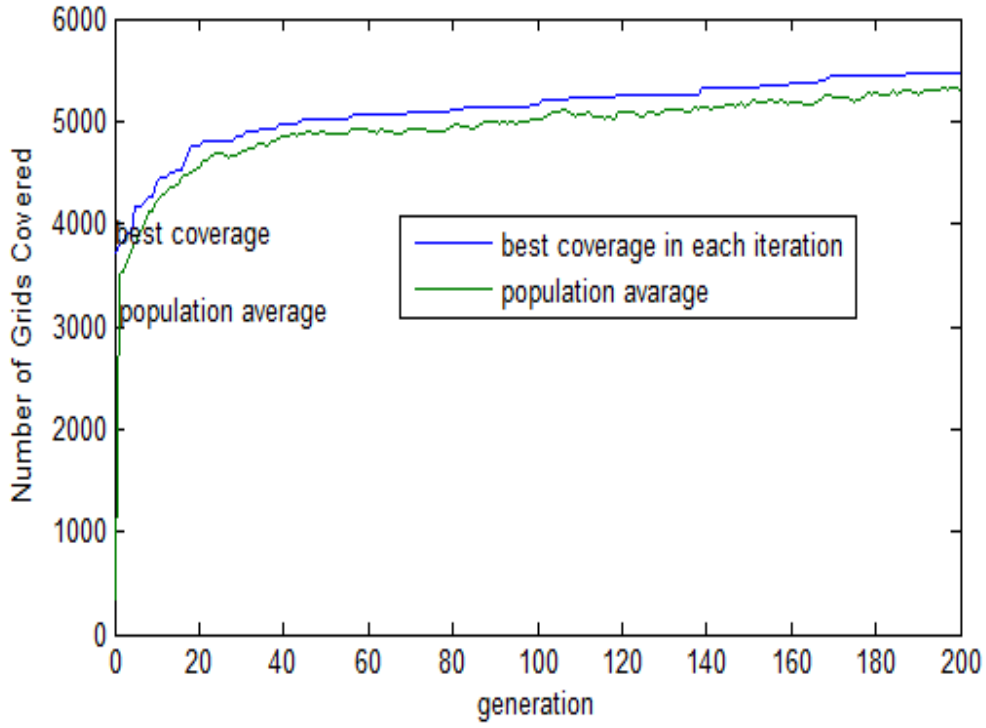

Figure 4.11 Hybrid Algorithm Result for Smooth Surface

The best and final area coverage out of 10,000 grids is 5468 grids. About 54.68% of the total area to be monitored is sensed as per the best solution of the Hybrid Algorithm. The result for smooth surface is better than the rough surface due to improved line of sight between node locations and the sensed grid. The smooth surface has improved about 268 grids when compared to the rough surface area coverage. This improvement is accounted totally for LOS improvement since the sensing range for both rough and smooth surface is the same as it is discussed in PSO results of rough and smooth surfaces under section 4.2.1 and 4.2.2. The improvement between rough and smooth surfaces in HA is about 5.15%. The sensed grids plot for smooth surface using HA found solution is shown the figure below.

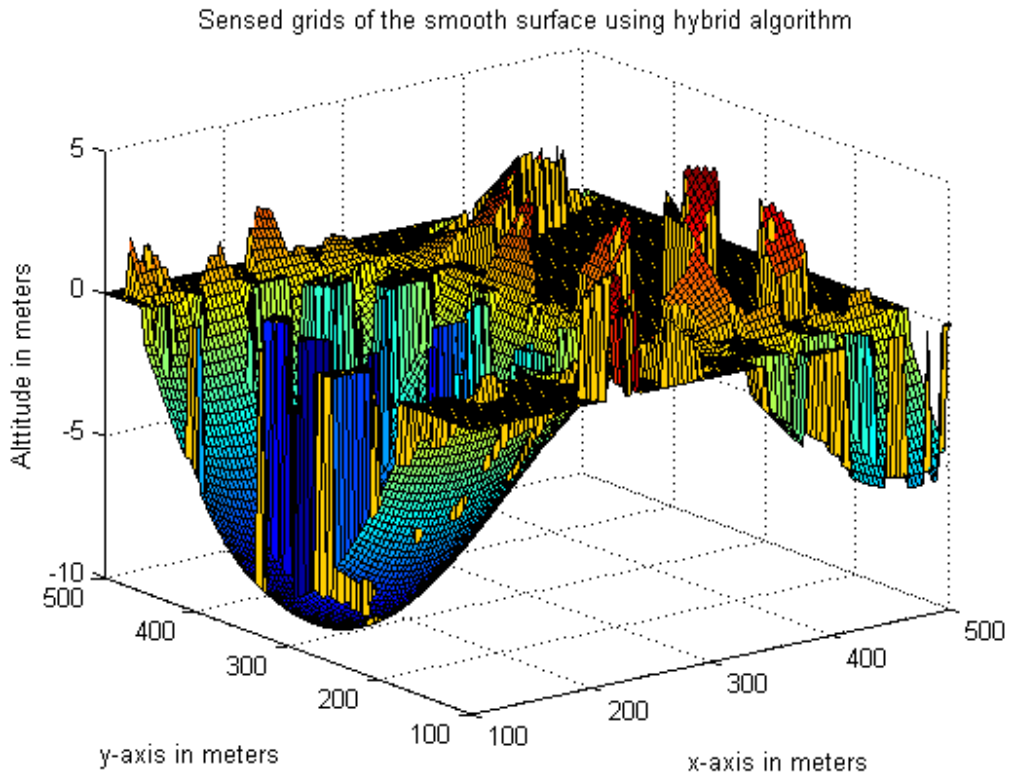

Figure 4.12 The plot of sensed grids of the smooth surface. (The black is not sensed and plot with other colors is sensed)

The un-sensed grids are multiplied by zero and looks like simple black color plane on above figure. The best positions for the sensor nodes of all 40s to be deployed on the smooth surface is given as per the result of the Hybrid algorithm as shown in the following table 4.6.

Node No.	X-Coordinate	Y-Coordinate	Z-Coordinate
1	108.8026	216.3592	-4.9483
2	109.4617	458.6208	-1.1185
3	114.0832	365.4701	-7.9192
4	122.1374	298.5308	-9.2812
5	133.329	421.0621	-4.6747
6	145.7383	180.784	-2.3332
7	166.0465	243.2197	-7.5571
8	181.1777	347.608	-9.1728
9	186.199	449.2735	-2.0872
10	186.6015	304.2647	-9.5206
11	194.9759	402.4281	-5.9002
12	206.9034	211.0537	-4.5143
13	232.0897	267.1404	-6.5223
14	234.9898	339.7018	-6.8846
15	236.3583	127.9734	2.0141
16	238.1996	464.9167	-0.4351
17	246.0301	169.7604	-0.7965
18	249.1379	387.2803	-4.5061
19	252.9913	197.5476	-2.2609
20	254.1656	446.761	-1.3682
21	262.9945	100.2213	2.6363
22	283.2763	287.8685	-2.935
23	284.4711	166.9725	-0.2889
24	287.1181	253.2697	-2.1912
25	292.0555	455.5186	-0.3432
26	296.8786	345.2383	-1.6371
27	316.1761	370.3141	0.1707
28	318.3962	149.3752	-0.0326
29	322.039	246.7498	0.615
30	336.2882	416.1503	1.1489
31	345.9147	167.2619	0.3174
32	356.9614	439.1883	1.3077
33	371.7754	209.6198	2.7326
34	376.6521	113.9692	-2.4447
35	384.0933	459.9608	0.7244
36	408.5898	171.6039	1.1725
37	416.3649	468.112	0.2667

38	430.582	130.7228	-2.3929
39	461.9824	497.5918	-2.5937
40	475.19	132.0745	-2.4726

Table 4.6 Node Positions found using HA for the typical smooth surface.

The HA outperformed the GA and very much exceeded the PSO when we compare the results of the three algorithms for the same smooth typical surface used. We have also used the same iteration number and all other parameters that can be shared by the three algorithms. HA has exceeded PSO by about 995 grids with a percentage growth of about 22.24%. When we also compare the results of HA and GA algorithms for the same smooth typical surface used and the same iteration number and all other parameters that can be shared by the two algorithms, HA has exceeded GA by about 181 grids with a percentage growth of about 3.42%.

4.5. Comparison of PSO, GA and Hybrid Algorithms

Since 3D surface is considered, coverage area matters more than coverage volume. But when the surface area is optimized (maximized), the coverage volume gets also maximized although the best optimal volume coverage cannot be achieved along with optimal surface coverage. We can see their comparisons on smooth and rough surface.

4.5.1. Comparison of PSO, GA and Hybrid on Rough Surface

In PSO algorithm, the initial population is created randomly and the objective function is calculated. We have used PSO to generate the optimal solution for 200 iterations and population size of 20 to cover maximum area with minimum number of 40 nodes. This algorithm determines search path according to the velocity and current position of particle without more complex evolution operation. However, as we see the figure bellow PSO algorithm has less performance than GA and HA.

In all simulation results in our thesis, the values of the common parameters used in each algorithm such as population size, number of nodes and total iteration number were chosen to be the same. However, each algorithm has their own other specific parameters. As we can see the figure bellow GA algorithm has greater performance than PSO but has less performance than HA. The HA

outperformed the GA and very much exceeded the PSO when we compare the results of the three algorithms for the same rough typical surface used.

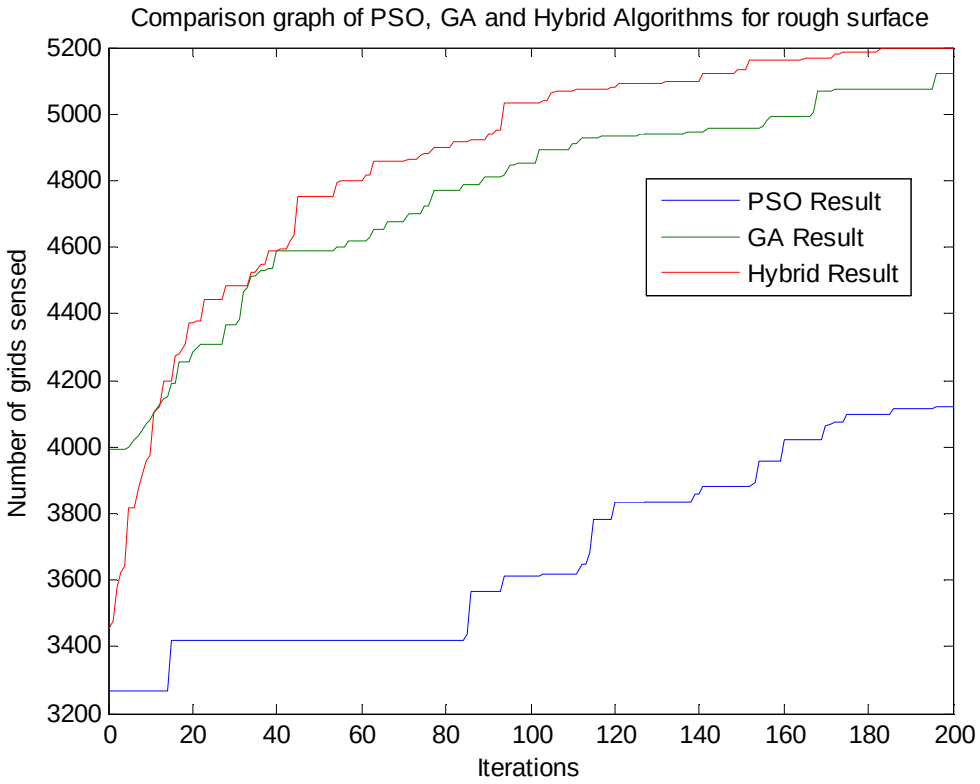

Figure 4.13 Comparison of PSO, GA and HA for the rough surface

4.5.2. Comparison of PSO, GA and Hybrid on Smooth Surface

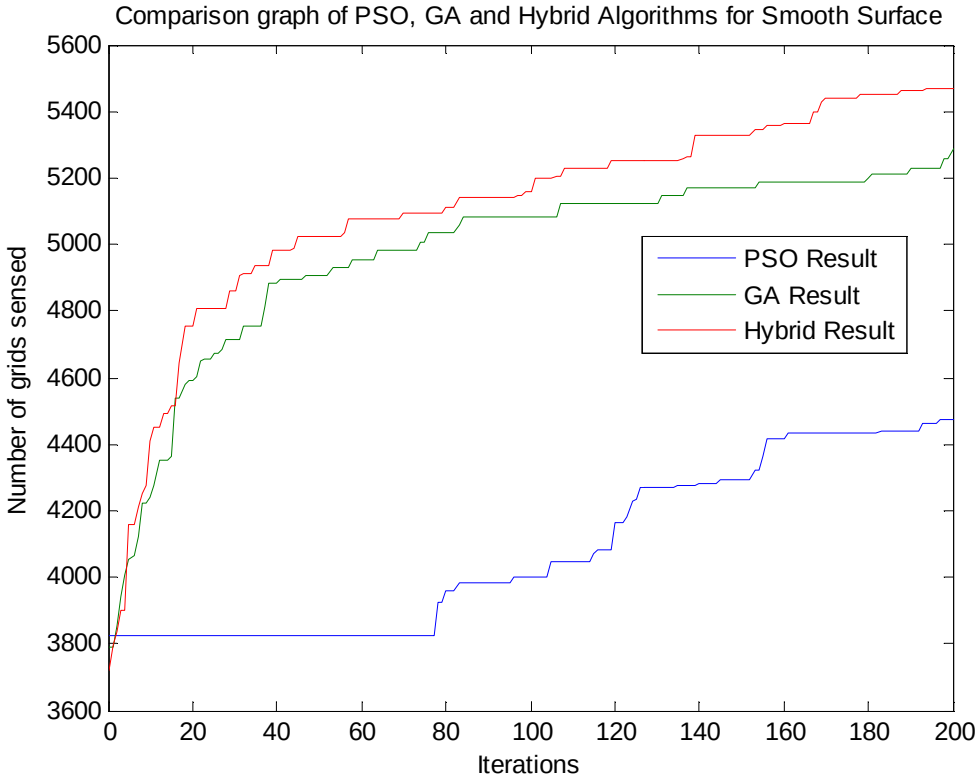

Figure 4.14 Comparison result of PSO, GA and HA on smooth surface

The figure above in figure 4.14 presents a comparison of PSO, GA and HA coverage area performance throughout 200 iterations. In all simulation results in this thesis, the values of the common parameters used in each algorithm such as population size, number of nodes and total iteration number were chosen to be the same. From the figure we can see that GA algorithm has greater performance than PSO but has less performance than HA. The HA outperformed the GA and very much exceeded the PSO when we compare the results of the three algorithms for the same smooth typical surface used.

5. Conclusion

5.1. Summary of Results and Conclusions

WSNs are composed of cooperating sensor nodes that gather information from the environment, which have been considered in many potential applications, such as surveillance, biological detection, Habitat and Ecosystem Monitoring, Monitoring Groundwater Contamination and Industrial Process Monitoring etc... One of the most critical issues of WSNs is the sensor deployment in order to achieve maximum coverage. In order to maximize coverage, sensors need to be placed in a position such that the sensing capability of the network is fully utilized to ensure high QoS. Coverage is one of the main problems in WSNs deployment. Previous research works on sensor deployment mainly focused on 2D plane or in 3D volume. But now a days, SD studies extended to 3D surfaces or terrain, to achieve the highest overall sensing quality. In our thesis work, we worked to develop an optimal WSNs deployment on 3D surfaces to maximize area coverage under constrained number of nodes.

Researchers have used different methods and algorithms to overcome coverage problem. Population-based optimization algorithms find near-optimal solutions to the difficult optimization inspired by natural probabilistic evolution. In our research work, GA and PSO are selected to form HA to find optimal locations of sensors based on a fitness function. We have selected the two algorithms to exploit the best features of the algorithms in combination.

To decide the type of surface, which use for simulating the deployment, we have considered function generated surface with close resemblance to the geographical surface. We have used two typical surfaces, rough and relatively smooth, to compare the results of the GA, PSO and our HA in the optimal deployment of sensors. For both typical rough and smooth surfaces, the fitness function used in the algorithms is calculated based on coverage of all sensors in the ROI. A simulating program for both surface types and all the three algorithms has been developed using MATLAB.

In comparison of simulation results of PSO and HA on the same rough typical surface, HA has exceeded PSO with a percentage of 26.12%. When we also compare the results of HA and GA algorithms for the same rough typical surface used, HA has exceeded GA with a percentage of 1.58%.

The comparisons of simulation results on typical smooth surface for all the three algorithms has also been made. The comparison shows that HA has exceeded PSO with a percentage growth of about 22.24%. In the same manner, HA has exceeded GA with a percentage of 3.42%.

Therefore, we found that the HA outperformed the PSO and GA algorithms with significant percentage for both rough and smooth terrain types. Next to the HA, GA has a very good performance when compared to PSO.

5.2. Future works

This section describes the possible extensions of research presented in our thesis.

- We assume that the mobility of nodes are static or stationary. This analysis can be extended assuming that the mobility of nodes are mobile.
- In surface modelling, in this thesis we have considered function generated surface with close resemblance to the geographical surface. We have planned to extend our research work by modelling the surface from Geographical data from Google Earth or satellite image and use the concept of image processing.
- Our research work has only considered coverage area for deployment. In future works, other metrics of deployment such as routing and efficient power management can also be taken into consideration in deployment of sensors.

Appendix: Sample Codes for Algorithms of 3-D Terrain

A. Code of PSO Algorithm for rough surface

```
function localpar = PSOB(NoNodes)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here
% Particle Swarm Optimization - PSO
% Haupt & Haupt
% 2003
% Initializing variables
popsize = 20; % Size of the swarm
npar = 2*NoNodes; % Dimension of the problem
maxit = 200; % Maximum number of iterations
c1 = 1; % cognitive parameter
c2 = 4-c1; % social parameter
C=1; % constriction factor
% Initializing swarm and velocities
para=rand(popsize*10,npar); % random
cost=costFunctionGA(para); % calculates population cost
% using ff
[cost,ind]=sort(cost); % min cost in element 1
para=para(ind,:); % sort continuous
par(1:20,:)=para(1:20,:); % random population of
% continuous values
%par=localpara;
vel = rand(popsize,npar); % random velocities
% Evaluate initial population
cost=costFunctionGA(par); % calculates population cost using
% ff
maxc(1)=max(cost); % max cost
meanc(1)=mean(cost); % mean cost
globalmax=maxc(1); % initialize global maximum
% Initialize local maximum for each particle
localpar = par; % location of local maxima
localcost = cost; % cost of local maxima
% Finding best particle in initial population
[globalcost,indx] = max(cost);
globalpar=par(indx,:);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Start iterations
iter = 0; % counter
while iter < maxit
iter = iter + 1;
% update velocity = vel
w=(maxit-iter)/maxit; %inertia weiindxht
r1 = rand(popsize,npar); % random numbers
r2 = rand(popsize,npar); % random numbers
vel = C*(w*vel + c1 *r1.*(localpar-par) +c2*r2.*(ones(popsize,1)*globalpar-
par));
% update particle positions
par = par + vel; % updates particle position
overlimit=par<=1;
underlimit=par>=0;
```

```
par=par.*overlimit+not(overlimit);
par=par.*underlimit;
% Evaluate the new swarm
cost = costFunctionGA(par); % evaluates cost of swarm
% Updating the best local position for each particle
bettercost = cost > localcost;
localcost = localcost.*not(bettercost) +cost.*bettercost;
localpar(find(bettercost),:) = par(find(bettercost),:);
% Updating index g
[temp, t] = max(localcost);
if temp>globalcost
globalpar=par(t,:); indx=t; globalcost=temp;
end
[iter globalpar globalcost] % print output each
% iteration
maxc(iter+1)=max(cost); % max for this
% iteration
globalmax(iter+1)=globalcost; % best max so far
meanc(iter+1)=mean(cost); % avg. cost for
% this iteration
end% while
figure(24)
iters=0:length(maxc)-1;
plot(iters,maxc,iters,meanc,iters,globalmax);
xlabel('generation');ylabel('cost');
text(0,maxc(1),'best');text(1,maxc(2),'population average')

end
```

B. Code of GA Algorithm for rough surface

```
function [par,minc] = GAMIN(NoNodes)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
% Continuous Genetic Algorithm
%
% minimizes the objective function designated in ff
% Before beginning, set all the parameters in parts
% I, II, and III
% Haupt & Haupt
% 2003
%_____________________________________________________________
% I Setup the GA
ff='Cost Function'; % objective function
npar=2*NoNodes; % number of optimization variables
varhi=1; varlo=0; % variable limits
% II Stopping criteria
maxit=200; % max number of iterations
mincost=-9999999; % minimum cost
%_____________________________________________________________
% III GA parameters
popsize=20; % set population size
mutrate=.02; % set mutation rate
selection=0.5; % fraction of population kept
Nt=npar; % continuous parameter GA Nt=#variables
keep=floor(selection*popsize); % #population
```

```
% members that survive
nmut=ceil((popsize-1)*Nt*mutrate); % total number of
% mutations
M=ceil((popsize-keep)/2); % number of matings
%_____________________________________________________
% Create the initial population
iga=0; % generation counter initialized
par=rand(popsize,npar); % random
cost=costFunctionGA(par); % calculates population cost
% using ff
[cost,ind]=sort(cost); % min cost in element 1
par=par(ind,:); % sort continuous
minc(1)=10000/min(cost); % minc contains min of
meanc(1)=1000/mean(cost); % meanc contains mean of population
%_____________________________________________________
% Iterate through generations
while iga<maxit
iga=iga+1; % increments generation counter
%_____________________________________________________
% Pair and mate
M=ceil((popsize-keep)/2); % number of matings
prob=flipud([1:keep]'/sum([1:keep])); % weights
% chromosomes
odds=[0 cumsum(prob(1:keep))']; % probability
% distribution
% function
pick1=rand(1,M); % mate #1
pick2=rand(1,M); % mate #2
% ma and pa contain the indicies of the chromosomes
% that will mate
ic=1;
while ic<=M
for id=2:keep+1
if pick1(ic)<=odds(id) & pick1(ic)>odds(id-1)
ma(ic)=id-1;
end
if pick2(ic)<=odds(id) & pick2(ic)>odds(id-1)
pa(ic)=id-1;
end
end
ic=ic+1;
end
%_____________________________________________________
% Performs mating using single point crossover
ix=1:2:keep; % index of mate #1
xp=ceil(rand(1,M)*Nt); % crossover point
r=rand(1,M); % mixing parameter
for ic=1:M
xy=par(ma(ic),xp(ic))-par(pa(ic),xp(ic)); % ma and pa
% mate
par(keep+ix(ic),:)=par(ma(ic),:); % 1st offspring
par(keep+ix(ic)+1,:)=par(pa(ic),:); % 2nd offspring
par(keep+ix(ic),xp(ic))=par(ma(ic),xp(ic))-r(ic).*xy;
% 1st
par(keep+ix(ic)+1,xp(ic))=par(pa(ic),xp(ic))+r(ic).*xy;
% 2nd
```

```
if xp(ic)<npar % crossover when last variable not selected
par(keep+ix(ic),:)=[par(keep+ix(ic),1:xp(ic)),par(keep+ix(ic)+1,xp(ic)+1:npar
)];
par(keep+ix(ic)+1,:)=[par(keep+ix(ic)+1,1:xp(ic)),par(keep+ix(ic),xp(ic)+1:np
ar)];
end % if
end
%_____________________________________________________
% Mutate the population
mrow=sort(ceil(rand(1,nmut)*(popsize-1))+1);
mcol=ceil(rand(1,nmut)*Nt);
for ii=1:nmut
par(mrow(ii),mcol(ii))=(varhi-varlo)*rand+varlo;
% mutation
end % ii
%_____________________________________________________
% The new offspring and mutated chromosomes are
% evaluated
cost=costFunctionGA(par);
%_____________________________________________________
% Sort the costs and associated parameters
[cost,ind]=sort(cost);
par=par(ind,:);
%_____________________________________________________
% Do statistics for a single nonaveraging run
minc(iga+1)=10000/min(cost);
meanc(iga+1)=10000/mean(cost);
%_____________________________________________________
% Stopping criteria
if iga>maxit | cost(1)<mincost
break
end
[iga cost(1)]
end %iga
%_____________________________________________________
% Displays the output
best=par(1,:);
day=clock;
disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
disp(['optimized function is ' ff])
format short g
disp(['popsize = ' num2str(popsize) ' mutrate = ' num2str(mutrate) ' # par =
' num2str(npar)])
disp(['#generations=' num2str(iga) ' best cost=' num2str(cost(1))])
disp(['best solution'])
disp([num2str(par(1,:))])
disp('continuous genetic algorithm')
figure(24)
iters=0:length(minc)-1;
plot(iters,minc,iters,meanc);
xlabel('generation');ylabel('cost');
text(0,minc(1),'best');text(1,minc(2),'population average')
end
```

C. Code of HPSOGA Algorithm for rough surface

```
function [par,minc] = GASMOD(NoNodes)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
% Continuous Genetic Algorithm
%
% minimizes the objective function designated in ff
% Before beginning, set all the parameters in parts
% I, II, and III
% Haupt & Haupt
% 2003
%_____________________________________________________________
% I Setup the GA
ff='Cost Function'; % objective function
npar=2*NoNodes; % number of optimization variables
varhi=1; varlo=0; % variable limits
% II Stopping criteria
maxit=200; % max number of iterations
mincost=-9999999; % minimum cost
%_____________________________________________________________
% III GA parameters
popsize=20; % set population size
mutrate=.02; % set mutation rate
selection=0.5; % fraction of population kept
Nt=npar; % continuous parameter GA Nt=#variables
keep=floor(selection*popsize); % #population
% members that survive
nmut=ceil((popsize-1)*Nt*mutrate); % total number of
% mutations
M=ceil((popsize-keep)/2); % number of matings
%_____________________________________________________________
% Create the initial population
iga=0; % generation counter initialized
par=rand(popsize,npar);
cost=costFunctionGA(par); % calculates population cost
% using ff
[cost,ind]=sort(cost); % min cost in element 1
par=par(ind,:); % sort continuous
minc(1)=10000/min(cost); % minc contains min of
meanc(1)=1000/mean(cost); % meanc contains mean of population
localpar = par; % location of local minima
localcost = cost; % cost of local minima
%_____________________________________________________________
% Iterate through generations
stagCounter=0;
gaCounter=0;
while iga<maxit
iga=iga+1; % increments generation counter
%_____________________________________________________________
% Pair and mate
M=ceil((popsize-keep)/2); % number of matings
prob=flipud([1:keep]'/sum([1:keep])); % weights
% chromosomes
odds=[0 cumsum(prob(1:keep))']; % probability
% distribution
```

```
% function
pick1=rand(1,M); % mate #1
pick2=rand(1,M); % mate #2

% Special pso operators to avoid possibe local minima traps

%////////////

% ma and pa contain the indicies of the chromosomes
% that will mate
ic=1;
while ic<=M
for id=2:keep+1
if pick1(ic)<=odds(id) & pick1(ic)>odds(id-1)
ma(ic)=id-1;
end
if pick2(ic)<=odds(id) & pick2(ic)>odds(id-1)
pa(ic)=id-1;
end
end
ic=ic+1;
end
%_____________________________________________________________
% Performs mating using single point crossover
ix=1:2:keep; % index of mate #1
xp=ceil(rand(1,M)*Nt); % crossover point
r=rand(1,M); % mixing parameter
for ic=1:M
xy=par(ma(ic),xp(ic))-par(pa(ic),xp(ic)); % ma and pa
% mate
par(keep+ix(ic),:)=par(ma(ic),:); % 1st offspring
par(keep+ix(ic)+1,:)=par(pa(ic),:); % 2nd offspring
par(keep+ix(ic),xp(ic))=par(ma(ic),xp(ic))-r(ic).*xy;
% 1st
par(keep+ix(ic)+1,xp(ic))=par(pa(ic),xp(ic))+r(ic).*xy;
% 2nd
if xp(ic)<npar % crossover when last variable not selected
par(keep+ix(ic),:)=[par(keep+ix(ic),1:xp(ic)),par(keep+ix(ic)+1,xp(ic)+1:npar
)];
par(keep+ix(ic)+1,:)=[par(keep+ix(ic)+1,1:xp(ic)),par(keep+ix(ic),xp(ic)+1:np
ar)];
end % if
end
%_____________________________________________________________
% Mutate the population
mrow=sort(ceil(rand(1,nmut)*(popsize-1))+1);
mcol=ceil(rand(1,nmut)*Nt);
for ii=1:nmut
par(mrow(ii),mcol(ii))=(varhi-varlo)*rand+varlo;
% mutation
end % ii
%_____________________________________________________________
% The new offspring and mutated chromosomes are
% evaluated
lastBestCost=cost(1);
cost=costFunctionGA(par);
```

```
%_____________________________________________________________
% Sort the costs and associated parameters
[cost,ind]=sort(cost);
par=par(ind,:);
%Embedding selection of PSO onto GA with some conditions
if(lastBestCost-cost(1)==0)
    stagCounter=stagCounter+1; %The satgCounter counts the number of
iterations without the best value with no change
else
    stagCounter=0;
end
if(stagCounter<2 && iga<100) %The PSO selction is embedded onto GA when there
is no stagnancy and iteration number <100
  [localcost,index]=sort(localcost);
  localpar=localpar(index,:);
  bettercost = cost < localcost;
  localcost = localcost.*not(bettercost) + cost.*bettercost;
  localpar(find(bettercost),:) = par(find(bettercost),:);
  par(2:2:10,:)=localpar(2:2:10,:); % Takes the cromosomes of the population
from the local best as 2,4,6,8,10
else
    localpar=par;
end
%_____________________________________________________________
% Do statistics for a single nonaveraging run
minc(iga+1)=10000/min(cost);
meanc(iga+1)=10000/mean(cost);
%_____________________________________________________________
% Stopping criteria
if iga>maxit | cost(1)<mincost
break
end
[iga cost(1)]
end %iga
%_____________________________________________________________
% Displays the output
best=par(1,:);
day=clock;
disp(datestr(datenum(day(1),day(2),day(3),day(4),day(5),day(6)),0))
disp(['optimized function is ' ff])
format short g
disp(['popsize = ' num2str(popsize) ' mutrate = ' num2str(mutrate) ' # par =
' num2str(npar)])
disp(['#generations=' num2str(iga) ' best cost=' num2str(cost(1))])
disp(['best solution'])
disp([num2str(par(1,:))])
disp('continuous genetic algorithm')
figure(24)
iters=0:length(minc)-1;
plot(iters,minc,iters,meanc);
xlabel('generation');ylabel('cost');
text(0,minc(1),'best');text(1,minc(2),'population average')
end
```

D: Rough surface maker function

```
[X,Y]=meshgrid(100:0.5:500);
Z=10*sin(0.02*X).*cos(0.04*Y);
mesh(X,Y,Z);
xlabel('x-axis in meters');
ylabel('y-axis in meters');
zlabel('z-axis (altitude) in meters');
```

E: Line of Sight function for Rough surface

```
function vis = los6(locpt,pt)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
if(pt~=locpt)
  A=pt-locpt;
  if(pt(1)<locpt(1))
      x=pt(1):0.5:locpt(1);
  elseif(pt(1)>locpt(1))
      x=locpt(1):0.5:pt(1);
  end

  y=A(2)/A(1)*(x-pt(1))+pt(2);
  z=A(3)/A(1)*(x-pt(1))+pt(3);
  surfz=10*sin(0.02*x).*cos(0.04*y);
  vis=1;
  for i=1:numel(z)
      if(surfz(i)>z(i))
          vis=0;
          break;
      end
  end
else
    vis=1;
end
end
```

F: Cost Function of 3-D Rough surface

```
function fitVal= costFunctionGA(Pop)
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here
matSize=size(Pop);
Pop=400*Pop+100;
fitVal=zeros(matSize(1),1);
[X,Y]=meshgrid(100:4:500);
Z=10*sin(0.02*X).*cos(0.04*Y);
loc=zeros(3,1);
for i=1:matSize(1)
    sol=Pop(i,:);
    A=zeros(numel(Z),1);
    for k=1:(numel(sol)/2)
        loc=sol(2*k-1:2*k);
        loc(3)=10*sin(0.02*loc(1)).*cos(0.04*loc(2));
        for j=1:numel(Z)
            pt=[X(j) Y(j) Z(j)];
            d=sqrt((loc(1)-pt(1)).^2+(loc(2)-pt(2)).^2+(loc(3)-pt(3)).^2);
            if(d<35)
                vis=los6(loc,pt);
                if(vis == 1)
                  A(j)=1;
                end
            end
        end
    end
    fitVal(i)=10000/numel(find(A));
end

end
```

References

[1] Vinod Kumer,Pravav Khanna and Sanjay Bisht, “Adaptive PSO based Algorithm for Optimal WSN Deployment in 3 Dimensional terrains”, International Journal of Computer applications, 2012.

[2] Nikitha Kukunuru,Babu Rao Thella and Rajya Lakshmi Davuluri, “Sensor Deployment using Particle Swarm Optimization”, International Journal of Engineering Science and Technology Vol.2(10), 2010.

[3] Miao Jin, Guodong Rong, Hongyi Wu, Liang Shuai, and Xiaohu Guo, “Optimal Surface Deployment Problem in Wireless Sensor Networks”, Center for Advanced Computer Studies, University of Louisiana Department of Computer Science, University of Texas, 2012.

[4] Numan Unaldi, Samil Temel, and Vijayan K. Asari, “ Method for Optimal Sensor Deployment on 3D Terrains Utilizing a Steady State Genetic Algorithm with a Guided Walk Mutation Operator Based on the Wavelet Transform”, Sensors, ISSN 1424-8220, **2012**.

[5] Raghavendra V. Kulkarni and Ganesh Kumar Venayagamoorthy, “*Bio*-inspired Algorithms for Autonomous Deployment and Localization of Sensor Nodes”, Senior Member, IEEE, 1094-6977, 2010.

[6] Wen-Hwa Liao,Yucheng Kao, Ru-Ting Wu, , “Ant colony optimization based sensor deployment protocol for wireless sensor networks”, Department of Information Management, Tatung University, Taipei, Taiwan

[7] Damien B. Jourdan, Olivier L. de Weck, “Layout Optimization for a Wireless Sensor Network Using a Multi-Objective Genetic Algorithm”, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology 77 Massachusetts Avenue, Cambridge, MA 02139, USA.

[8] Michał Marks, “A Survey of Multi-Objective Deployment in Wireless Sensor Networks”, Journal of Telecommunications and Information Technology, 2010.

[9] Elena Gaura, Lewis Girod, James Brusey, Michael Allen and Geoffrey Challen, “Wireless Sensor Networks”, Deployments and Design Frameworks.

[10] Azzedine Boukerche, “Algorithms and Protocols for Wireless Sensor Networks”, John Wiley & Sons, Inc., 2009.

[11] Naeim Rahmani, Farhad Nematy, Amir Masoud Rahmani and Mehdi Hosseinzadeh, “Node Placement for Maximum Coverage Based on Voronoi Diagram Using Genetic Algorithm in Wireless Sensor Networks”, Australian Journal of Basic and Applied Sciences, 5(12): 3221-3232, 2011 ISSN 1991-8178

[12] Wint Yi Poe and Jens B. Schmitt, “Node Deployment in Large Wireless Sensor Networks: Coverage, Energy Consumption, and Worst-Case Delay”, *AINTEC’09,* November 18–20, 2009,

[13] You-Chiun Wang, Chun-Chi Hu, and Yu-Chee Tseng, “Efficient Deployment Algorithms for Ensuring Coverage and Connectivity of Wireless Sensor Networks”, Department of Computer Science and Information Engineering National Chiao Tung University, Hsin-Chu, 30010, Taiwan.

[14] Sartaj Sahni and Xiaochun Xu, “Algorithms For Wireless Sensor Networks”, Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611 September 7, 2004

[15] Dervis Karaboga and Bahriye Akay, “A comparative study of Artificial Bee Colony algorithm”, Erciyes University, The Department of Computer Engineering, Melikgazi, 38039 Kayseri, Turkey

[16] Gaojuan Fan , Ruchuan Wang, Haiping Huang, Lijuan Sun and Chao Sha, “Coverage-Guaranteed Sensor Node Deployment Strategies for Wireless Sensor Networks”, *Sensors* vol. *10*, 2010

[17] Shahriar Asta1 and A. Şima Uyar, “A Novel Particle Swarm Optimization Algorithm”,Computer & Informatics Faculty Istanbul Technical University, Istanbul, Turkey

[18] Chia-Feng Juang, *Member, IEEE,* "A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design", IEEE Transactions on Systems, man, and Cybernetics—Part B: Cybernetics, VOL. 34, NO. 2, April 2004.

[19] Naser Zamanan , Jan K. Sykulski and A. K. Al-Othman, "Real Coded Genetic Algorithm Compared to the Classical Method of Fast Fourier Transform in Harmonics Analysis**",**Sept.2006.

[20] Dr.Omari Mohammed, "Introduction to Genetic Algorithms", from Support de Course, Springer 2008.

[21] Randy L. Haupt and Sue Ellen Haupt, Practical Genetic Algorithms, Second Edition, John Wiley & Sons, Inc, 2004.

[22] Gaige Wang, Lihong Guo ,Hong Duan , Luo Liu and Heqi Wang, **"**Dynamic Deployment of Wireless Sensor Networks by Biogeography Based Optimization Algorithm", Journal of Sensor and Actuator Networks ISSN 2224-2708,2012.

[23] Chi-Fu Huang and Yu-Chee Tseng, "The Coverage Problem in a Wireless Sensor Network", Mobile Networks and Applications 10, 519–528, 2005.

[24] Jianguo Shi and Changjie Zhou, "Two-Stage Dynamic Sensor Deployment Strategy Based on Virtual Force and Genetic Algorithm in Wireless Sensor Networks", *I.J.* Education and Management Engineering 2012.

[25] Xiaole Bai,Dong Xuan, Ziqiu, Yun, Ten H. Lai and Weijia Jia, "Complete Optimal Deployment Patterns for Full-Coverage and k-Connectivity (k =< 6) Wireless Sensor Networks", *MobiHoc'08,* May 26–30, 2008, Hong Kong SAR, China.Copyright 2008 ACM978-1-60558-073-9/08/05.