INCREASING LOCALIZATION PRECISION IN SENSOR NETWORKS WITH

MOBILE BEACONS – A GENETIC PATH PLANNING APPROACH

by

AWAIS IQBAL

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

DECEMBER 2008

ACKNOWLEDGEMENTS

First and foremost, I would like to thank the Almighty for his guidance and all his blessings.

I would like to express my gratitude to my supervisor, professor and great mentor Dr. Gergely Záruba, whose knowledge, expertise and patience added a lot to my experience. I wouldn't have done all this if I didn't have his support and for this I am truly and sincerely thankful to him. I would also like to thank my committee members Dr. Manfred Huber and Dr. David Levine for all their help and support. I am honored and proud to be mentored by them.

I would also like to thank my family for their support in every step of my life. I am grateful to my brother for his help, my sister for telling me consistently 'You can do it' and to my parents for everything that I am today and the encouragement to make it all possible.

I must all also acknowledge Bito Irie of the Computer Science Department for all his help in time when I needed it most.

November 1, 2008

ABSTRACT

INCREASING LOCALIZATION PRECISION IN SENSOR NETWORKS WITH MOBILE BEACONS – A GENETIC PATH PLANNING APPROACH

Awais Iqbal, M.S

The University of Texas at Arlington, 2008

Supervising Professor: Dr. Gergely Záruba

This work describes a genetic algorithm based approach to approximate an optimal path for a mobile beacon node in a grid of stationary wireless sensors. As the beacon moves over the field of sensors it broadcasts its location. Sensors that are currently in the proximity of the beacon will receive this communication and can then use several of these messages to compute estimates on their locations. An optimal path is defined as a path that will result in the highest overall precision of location estimates among sensors given a maximum path length for the beacon. We assume that sensors are uniformly deployed in a predefined deployment area. We evaluate location precision calculating the maximum achievable accuracy using Cramer Rao Bound (CRB) for unbiased evaluation.

We describe the path of the mobile beacon using strings of ‘X’ and ‘Y’ coordinate pairs. As paths are described using strings, they lend themselves to genetic algorithm manipulations. Thus, to improve on the localization precision given by the path of the mobile beacon, a genetic optimization approach is used. Multiple genetic operators including mutation, splicing, selection and cross-over are used to create new paths which are evaluated for precision. Details of the genetic optimization approach to find better and better generations of paths are given. Extensive optimization simulations are performed in order to look for paths resulting in high precision. We describe the best paths found as well as look at the relationship of maximum path length versus precision of overall location estimates.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

Wireless sensor networks (WSNs) have received considerable attention in the recent years because they have the potential to revolutionize many segments of our life, from home health care through security to asset management. They are also expected to act as an enabler for pervasive computing environments. WSNs have applications in several military and civilian domains and thus have attracted a significant amount of research interest. WSNs are often viewed as wireless ad hoc networks where battery life and thus energy consumption is of foremost importance. WSNs nodes can be considered as small, simple computers that perform basic operations with unique characteristics such as limited power, possible mobility, unattended operation etc. They operate in a cooperative and distributed manned and often use more powerful central base stations (sink) to which they relay data and let these sinks perform extensive calculations.

Localization is the ability of a wireless sensor node to find out its own physical position within a network. This is an important and fundamental ability for embedded networks as such knowledge is essential in several key applications of WSNs. Generally, for a sensor node to obtain its location, it needs to be equipped with some sensor that can be used to for inferring such information. The most straight forward sensor for such applications can be a global positioning system (GPS) receiver. However, do to the size, price, energy consumption, and coverage problems of GPS receivers, it is often not feasible to equip simple sensor nodes with them. Thus, there were several techniques proposed for sensors to obtain estimates on their locations using only their mandatory radio transceivers. In these cases various techniques to communicate with other sensors (possibly with more elaborate sensors that have better estimates, e.g., through using a GPS receiver) in the WSN and using communication related sensory information (e.g., received signal strength, or time of flight) are used to obtain location estimates. WSN localization is thus an exciting field, with new algorithms and applications being developed at a fast pace.

1.1 Motivation

Location awareness is very important for wireless sensor networks since many applications like asset tracking or monitoring depend on knowing sensor nodes' physical location. Localization is also emerging as an important task for application, transport, network and data-link layers. The localization problem has been studied for several years and researchers have shown that accurate, but even semi accurate location awareness can help a significantly in improving such tasks as routing, data aggregation, energy conservation, path planning, and security [2, 10, 13].

Localization is a very challenging problem when it comes to wireless sensor networks. Although several different direct solutions have been proposed (e.g., using GPS), accurate localization can be very costly in terms of computational and energy consumption needs. This makes such approaches very costly and in some cases infeasible for inexpensive embedded devices. There is fine balance that needs to be achieved in terms of localization accuracy and the goal that is trying to be achieved from location awareness. Most of the time this balance can greatly vary with the application for which the WSN is designed. Thus, techniques that use only the wireless transceiver and basic information that this transceiver provides are highly popular in wireless sensor networks. In most of these approaches, so called *anchor nodes* are spread among regular sensor nodes that can directly determine their location (e.g., by having it hard coded at the time of placement or using GPS). Other sensor nodes will use (possibly multi-hop) wireless communication to estimate their location based on the locations of anchors with which they directly or indirectly communicate. In another approach, a mobile beacon is moved over the field, periodically transmitting its own location. Multiple communications with such mobile beacon can be used in sensor nodes to infer their own locations. Such mobile beacons can be represented by mobile robots that either use aerial or ground movement. The trajectory that such a mobile beacon travels has a great influence on the precision of the location estimates in the sensor nodes. In this paper we investigate how such trajectories or paths can be engineered off-line in order to reach the highest overall location estimate precision in the sensor nodes. Thus, *path planning* that we will be considering in this paper is different from the path planning problem usually addressed in robotics. Path planning in robotics deals with navigating in (a possibly unknown) an environment towards a certain goal. The problem we are discussing in this work, although somewhat related, is ultimately different.

1.2 Objective

In this thesis we will be discussing the path planning problem for a mobile beacon in a grid of stationary sensors. We will be focusing on the off-line problem, where the mobile beacon cannot change and reason about its path based on communications during localization, but has to travel a precalculated route or path over the network. In our work, sensors are assumed to be uniform randomly deployed in a predefined area. We will assume that we don't know the exact location of these sensors. The objective is to design a path for the mobile beacon that will try to communicate with as many sensors as possible thus trying to minimize localization error and thus improving the location accuracy of the sensor. Since the path of the mobile node is defined before the beacon starts moving we can also use the term static or off-line path planning. In order to restrict the time or energy the mobile beacon can spend during this localization phase (to make the problem more realistic), we will impose constraints on the total distance of the paths travelled.

We describe the path of the mobile beacon using strings of 'X' and 'Y' co-ordinate pairs. As paths are described using such strings, they lend themselves to genetic algorithm manipulations. Thus, to improve on the localization precision given by the path of the mobile beacon, a genetic optimization approach is used. Multiple genetic operators including mutation, splicing, selection and cross-over are used to create new paths which are evaluated for precision. Genetic algorithms provide a search strategy based on an evolutionary model. Previous research shows that this approach is a feasible heuristics when solving optimization problems [4]. An optimal path is defined as a path that will result in the highest overall precision of location estimates among sensors given a maximum path length for the beacon. This paper discusses the optimal path planning problem in two dimensions for easy understanding, however, it can be relatively easily scaled to three dimensional spaces.

1.3 Related Work

Wireless sensor networks (WSNs) are a special case of MANETs thus several localization algorithms that are created for MANETs also work in WSNs. However, as discussed previously WSNs have certain unique characteristics such as limited computational capacity and strict security requirements that make these algorithms very hard to implement.

Another solution that has been proposed for localization is based on semi definite programming (SDP). In such solutions certain is data is collected from the sensors, relayed to a sink, and solutions are computed at a centralized location using optimization approaches. These kinds of algorithms generally give the best result (given the same sensory reading) since they have the entire data set available. However, such centralized solutions are slightly difficult to implement in wireless sensor networks because of their highly distributed behavior and the fact that information related to position is not easily collected if no additional infrastructure is available.

For distributed location calculations, the presence of anchor nodes deployed over the sensor network are assumed. Another and arguably less expensive approach is to employ a mobile beacon as described above. There has been very limited work done previously on the path planning problem for mobile beacons in this kind in wireless sensor networks. This paper is based on the mobile beacon idea presented by Sichitiu and Ramadurai in [7] and Koutsonikolas, Das and Hu in [3]. Even though no significant solutions were presented in [7], the approach of the mobile beacon was interesting and attracted a lot of research interest. More work was conducted on the off-line mobile beacon path planning in [3, 12] where the author presented different types of static paths in relationship to localization.

In this paper, we extend the work the presented in [5]. The authors of [5] empirically engineer static paths (such as spirals, zigzags, and random walks) for mobile beacons and determine their feasibility in terms of total path travelled and overall localization precision. Just like in this work, the authors in [5] employ Cramer-Rao bounds (CRB) to estimate localization accuracy. CRB gives a lower bound on best localization error achievable which is a fair method for comparing different localization algorithms as it eliminates any biases (more information about Cramer-Rao bounds are given in chapter 3). Here we make one step further and try to automatically design and improve paths genetic algorithm optimization. Thus, our approach do not rely on a particular ad hoc static path strategy like previous research work but will investigate how better and better paths can be engineered automatically. Static path approaches presented in previous research have deal with several problems. One problem is that the path covers the entire area and doesn't take in to account that sensors might be deployed in one specific area. Genetic path planning on the other hand can be adapted to various sensor field set ups and sensor node distributions. Secondly, previous static path planning approaches presented different movement strategies and often overlooked certain constraints such as maximum distance traveled.

1.4 Thesis Organization

This thesis is contains seven chapters; the rest of the thesis is organized as follows. Chapter 2 defines how localization is achieved in wireless sensor networks and explains the basics of how the approach we propose in this paper works. Chapter 3 explains how Cramer Rao bounds can help in determining the optimal path by giving an unbiased estimation. Chapter 4 describes our specifically tailored genetic algorithm and different genetic operators that are used. It also explains how each of these operators can help obtaining a better path in each generation. Implementation of the approach is discussed in Chapter 5 while Chapter 6 presents the results followed by conclusion and future work in chapter 7.

CHAPTER 2

LOCALIZATION

As discussed earlier *localization* is the process/ability of a node to estimate its location in an environment. Location of nodes can be estimated in several ways. Localization techniques in wireless sensor networks typically require some form of communication between reference point and receiver. Distributed localization methods do not require centralized systems but rely on each node determining its own location with only limited communication with nearby nodes. Communication technologies used could be RF-based, acoustic based, etc. Distance of the receiver is calculated using simple geometric equations.

To understand localization we provide a simple example in which we describe a technique called trilateration [1]. To make the example simple we are going to use a two dimensional target area noting that the technique is easily extended to three dimensional space. Trilateration requires distance measurements to three known locations. As each distance measure creates a circle around the specific known location, if the distances are exact, then these circles intersect in exactly one point, providing the location of the target. This is depicted in Figure 2.1. Similar model is used by modern Global Positioning System (GPS) devices to calculate position of the receiver where known point are satellites and distances are measured by measuring time of flight information from each (well-synchronized) satellite.. Modern day GPS receivers often use more than three satellites to calculate the position of the receiver. This is called multilateration. However, GPS receivers come with a cost. They are often too expensive and/or power hungry to be used in low end mobile devices on mass scale. This has prompted researchers to look in to different methods that use the available limited resources effectively without compromising much on localization accuracy.

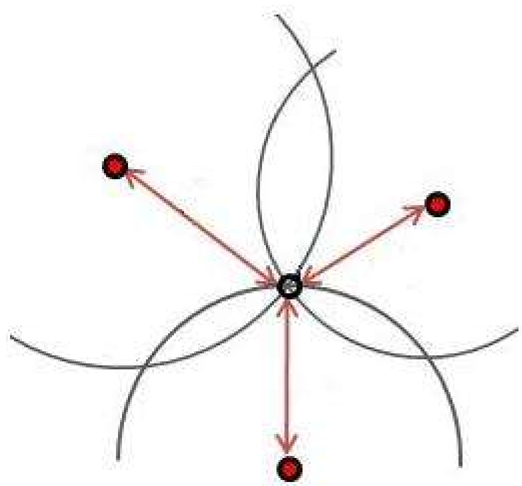


Figure 2.1: Localization using trilateration

Many available ranging methods require proprietary techniques and either require the use of additional equipment or rely on some specific environmental characteristic. GPS is one of them but there are other methods for location estimation, e.g., RADAR (Radio Detection and Ranging), LASER (Light Amplification by

Simulated Emission of Radiation) range finding and SONAR (Sound Navigation and Ranging). These methods are used widely in industrial and commercial equipment to estimate distances. However, these methods usually require line of sight and are power hungry, expensive, and bulky.

In sensor networks localization can be achieved by estimating distances between nodes and then perform computations on these data. In [12], the authors classify localization techniques into two categories: range-based and range-free. Range-based techniques rely on information determined from the communication between sensor nodes such as distance or angle and use that to compute the position of receiver. Range-based approaches have exploited several methods to measure relative location based information, such as time of arrival (TOA), received signal strength (RSSI), time difference of arrival (TDOA) and angle of arrival (AOA). Range-free methods are characterized by devices not measuring relative locations but do calculations solely based on the connectivity graph of the network (i.e., hop counting). I n general it is expected that range-based techniques provide better result than range-free techniques. However, range-based techniques can lead to wrong estimates if the relative location sensors are noisy or unreliable or no filtering on them is performed.

The recent boom in wireless networking has created sever inexpensive techniques for digital wireless transceivers, including IEEE 802.11, IEEE 802.15, or Bluetooth. As ranging information can be directly inferred from radio signal characteristics, localization based on just radio communications between sensor nodes. Techniques that use receive signal strength measurements are becoming popular as a basis for range-based localization in WSNs

Usually, localization methods in WSNs require that the sensor network is interspersed with a few nodes that can directly estimate their own location (e.g., using GPS). These nodes are referred to as anchor nodes. Another method that does not require the presence of such anchor nodes is to employ a mobile beacon that traverses the sensor network's coverage area and broadcasts it location to the sensor nodes. Nodes can then calculate their locations based on several such communications with the mobile beacon. This is depicted in Figure 2.2 where R1, R2, R3 and R4 represent different readings at different time intervals as the mobile beacon enter and leaves the range of the sensor.

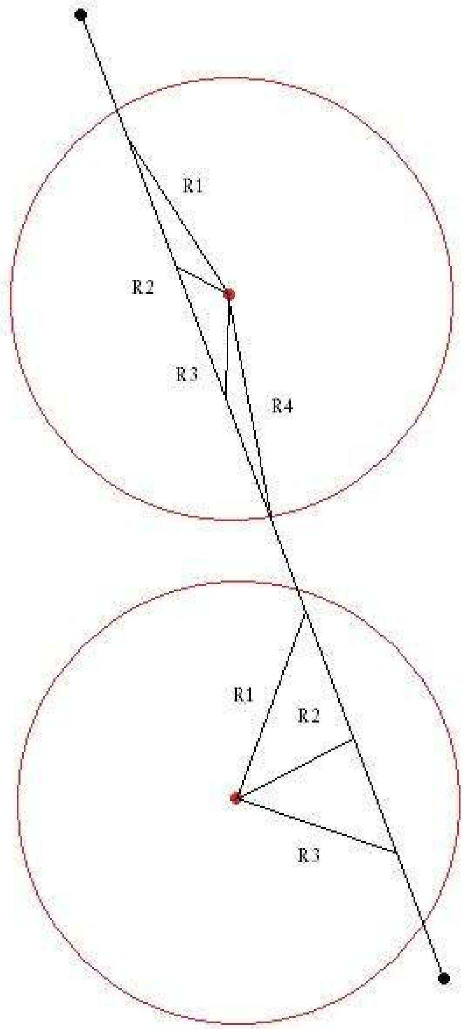


Figure 2.2: Readings from mobile node

Another interesting method is present in [7] where sensor location is viewed as probabilistic distribution over the defined area. The sensors take several RSSI readings from the mobile beacon at regular intervals and try to localize themselves. Any sensor node that is receiving a packet can infer that is somewhere around mobile node with a certain probability. This information constrains the possible locations of the sensor. Each sensor starts with a uniform distribution covering the entire area. As the mobile beacon passes through, distribution is updated to match the RSSI readings. Each node estimates its position corresponding to the RSSI measurement and the location of the mobile node including in the transmission.

In this paper we will be presenting a new approach to increase localization precision in sensor network using mobile beacons. Instead of relying on one particular path or pattern of paths we will present a solution that will adapt and find a path that is suited for the distribution of sensors. We start by uniformly deploying sensors in an area (see figure 2.3). These sensors are low powered sensor capable of handling only basic operations. These sensors have a certain communication range in which they can communicate with other sensors. Without the loss of generality we will assume that all sensors and the mobile node have the same range. It can be easily changed to implement a case where ranges are different. Sensors can communicate with each other when they are in communication range of the other sensor. Initially these sensors are not aware of their location. Equipping all sensors with GPS devices is not a good idea because of the power and/or computational requirements are not suitable for such devices. A slightly more powerful mobile beacon equipped with a GPS then moves in the defined area, broadcasting its location to nearby nodes at regular intervals. Nodes that are in range collect these readings and try to find their own position relative to the mobile beacon using RSSI. The black solid line in Figure 2.3 represents the path of the mobile node while the red slide represents the range of the sensor node. (To reduce complexity of the picture where overlapping ranges would be a criss-cross of circles, we elected to show smaller circles depicting a fraction of the range of the sensor node.) As the mobile node enters the communication range of the sensor several messages are passed back and forth on regular intervals (Figure 2.2).

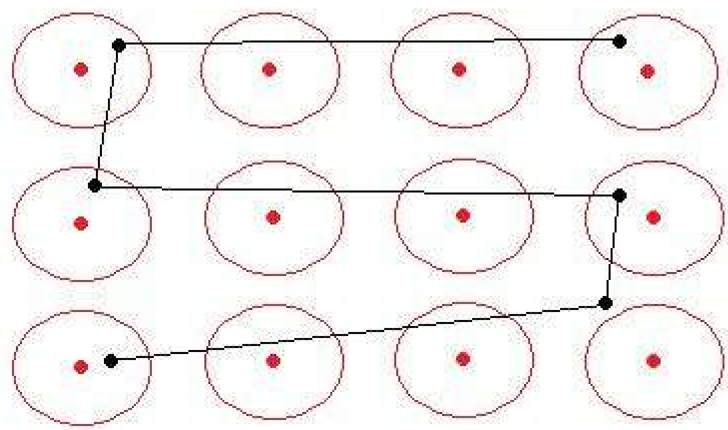


Figure 2.3: Example of mobile beacon trajectory

There is however one problem with mobile node traveling in a straight line. If a mobile nodes passes close to the sensor in a straight line, the proposed algorithm will not be able determine which side of the mobile node the sensor is. It is equally possible that the position is on either side of the line. To eliminate one of the candidates, one non-collinear packet must be received. For a sensor node to accurately localize itself, path must be decided in a way that each node receives at least three non-collinear readings from the mobile beacon.

CHAPTER 3

CRAMER RAO BOUND

In this paper we will be using Cramer Rao Bound (CRB) as an unbiased evaluator to paths. Since we focus on unbiasedness, we look for an estimator with the smallest possible variance. CRB will give the minimal achievable variance for any unbiased estimator that uses noisy measurements such as received signal strength (RSSI), time of arrival (TOA) and angle of arrival (AOA). CRB tells us how good a localization algorithm can do given a particular network, measurement type and noise scenario. The CRBs of individual measurement types under some common noise models are discussed in [9].

For our particular model where localization is performed using a mobile beacon, CRB values can be influenced by three factors. First factor is path length. A longer path means that there is a higher chance that sensor node will get more readings from the mobile node thus, resulting in better overall localization. Shorter paths on the other hand will result in less overall readings and thus comparatively inaccurate localization. This can be seen in Figure 3.2. Second factor that influences the CRB directly is broadcast interval. Looking at Figure 2.2 again, it can be seen that only 4 locations were broadcast. If this broadcast interval is reduced to half, the sensor will be able to receive at least 8 location packets which mean that it has more data for calculating its own location estimate. Therefore we can expect a better overall localization with a shorter broadcast interval. The last factor that is equally important is the broadcast range of the sensor and the mobile. Large range means that it is more likely that sensors will be in range of the mobile node which will allow each broadcast to cover more sensors. Therefore, we expect better overall localization with larger broadcast range. This can be seen in figure 3.2.

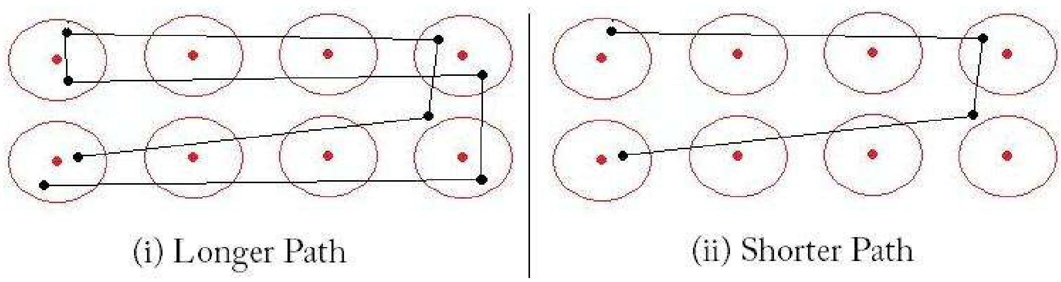


Figure 3.1: Beacon path length

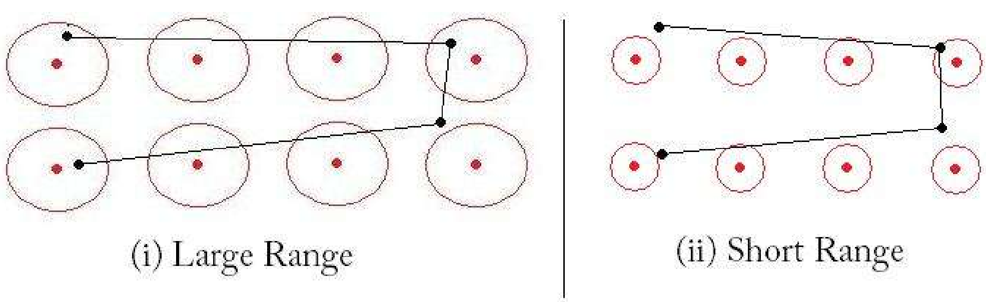


Figure 3.2: Node broadcast range

CHAPTER 4

GENETIC ALGORITHM

Genetic Algorithms (GA) are a heuristic search method based on Darwinian principles of natural selection. It is a mathematical programming technique that mimics biological evolution as a problem-solving strategy. Genetic algorithms are a particular class of biologically inspired algorithms using concepts such as populations, inheritance, mutation, selection, and crossover.

Mathematical optimization problems are powerful tool that can be used in computer science to search for solutions to a problem. The problem is modeled by an objective function (also called a fitness function) that allows solutions to the problem, to be evaluated. The task is to find the best solution, i.e., the solution with the highest (or depending on the problem the lowest) objective function value. In addition to the objective function, the problem can have constraints on solutions that need to be satisfied before the solutions fitness is evaluated. Solutions that satisfy all constraints are called feasible solutions. The fitness function is always problem dependent. For example if we take the knapsack problem where we want to maximize the total value of objects that we can put in a knapsack of some fixed capacity. The fitness of the solution is the sum of values of all objects in the knapsack if the selection is valid or zero otherwise (here the constraints are built into the objective function). For several categories of problems (i.e., categories of objective functions) there are simple programming methods to solve the problem. However, most objective functions and/or constraint sets require heuristics for finding semi-optimal solutions especially if the computational complexity of these problems is NP-hard.

Genetic algorithms (GA) represent one such heuristic solution technique. In GA we create a large set of feasible solutions and call them a population. Solutions should be represented in forms of strings. Each solution can be viewed as the genetic code of an individual. Each individual in the population is then evaluated by the objective function and the best performing individuals (see section below on how this is determined) are kept for mutations and breeding (crossover) to create a new population (the rest of the population is discarded). Through many generations, the "natural selection process" is going to result in better and better generations.

In our case, we represent paths that the mobile beacon could take by a string of x 'X" and 'Y' co-ordinate pairs. Thus these strings encode the "genetic information" of the paths. The genetic algorithm is run on an original feasible set of solutions, and by mutating and combining these solutions will result in better and better paths. Feasibility of the paths is determined by the total distance travelled. Fitness of individual paths is determined by an algorithmic calculation involving Cramer Rao bounds over the path.

The expectation is that the average fitness of the best members of the population will increase each round, and so by repeating this process for a given time, very good solutions to the problem can be discovered. A pseudo code to a genetic algorithm is given in appendix A of this work. A genetic algorithm consists of three major parts:

- Methods of selecting the survivor group (the parents to the next generation)
- Methods of selecting the genetic operator for creating each new individual
- And terminating condition.

4.1 Selection of Survival Group

There are many different techniques that a genetic algorithm can use to select candidates that will be kept for next generation. Some of these techniques are listed below.

- Most Fit (Elite): Only the best candidates from each generation are kept.
- Fitness-Proportionate: Most fit candidates are likely to be kept but not with certainty.
- Generational: Only the newest generation is kept:
- Roulette Wheel: Similar to Fitness proportionate but with chance of being kept proportional to the fitness level.

For our experiment we will keep only the most fit candidates from each generation. These candidates will not only be used in the next generation to produce more candidates but will be evaluated again themselves against the new individuals. Hence if none of the variations performed of the last generation produces a better individual, then the most fit candidates from the previous generation will be kept and used again (discarding all the newly created candidates). Thus the average fitness over the parent set is a monotone (but not strictly monotone) function over time.

4.2 Genetic Operators

Once selection of parents has happened, the parents will need to spawn offspring for new generations. There are two basic genetic operators that are used in GAs for creating offspring. These operators are crossover (breeding) and mutation. There are many ways these operators can be implemented. Arguably, the exact way the operators work should be a function of the problem itself. Thus in this work we will be presenting some custom mutation and crossover operators that try to achieve better path candidates for the mobile beacon.

4.2.1 Mutation

Mutation is the most powerful genetic operator. With mutation, populations of individuals can adapt very quickly, allowing small evolutions of good solutions to produce even better solutions. Mutation ensures a more complete coverage of the search-space by randomly altering genomes of individuals; in our case points along the path of the mobile beacon. Figure 4.1 provides an example of mutation, where one point of the path is moved from one location to another, resulting in all sensors being covered. This would likely result in a better overall localization precision. In our experiments a random path is selected from the elite (parent) paths and random points are modified. New individuals are admitted to the new population if their overall path length does not violate the maximum path length allowed.

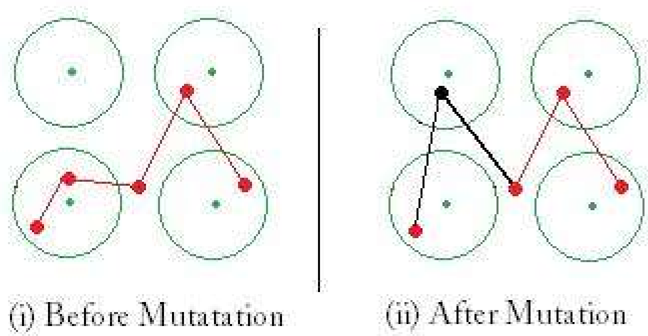


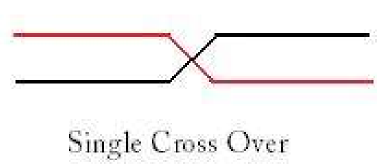
Figure 4.1: Mutation

4.2.2 Crossover

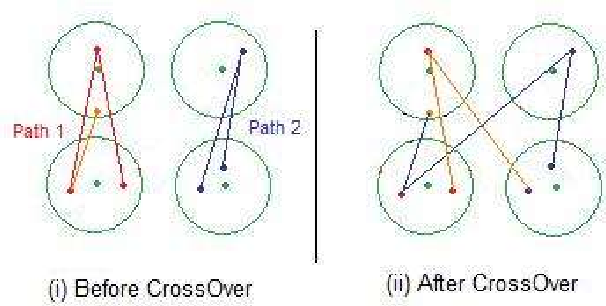
In GAs crossover is the means by which individuals exchange genetic material. In our case different paths are combined to create new paths. The most common type of crossover techniques used are single point and double point cross over.

4.2.2.1 Single Point Crossover

In single point crossover, two paths are randomly chosen from elite paths of last generations. Then, a random point is chosen on the individual paths, and path information is exchanged around this point. This means that all the path coordinates of the first path after this point are moved onto the second path and all the points of the second path after this point are moved to the first. This is extremely helpful in a scenario where two paths are getting local minimums in two different areas. When these two paths are combined, it can happen that a path will be created which provides more accurate localization. Consider Figure 4.2 (b). Individually both path 1 and path 2 are not good paths however; when these two paths are crossed they produce two much better paths.



(a)

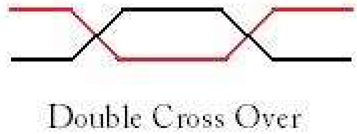


(b)

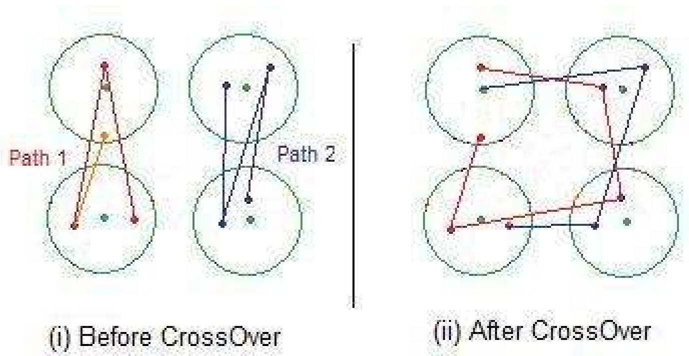
Figure 4.2: Single Point Cross Over

4.2.2.2 Double Point Crossover

Double point crossover works essentially the same way as single point; however, instead of swapping once, it swaps twice. Consider figure 4.3. First two paths are selected from the best paths of last generation. Then two random points are chosen in those paths and their contents are swapped between those two points. This is very helpful in case where sensors are divided in two different areas and each path is only covering only one area. It is worth emphasizing that double point crossover is more disruptive than single point crossover in the sense that it recombines the genetic material more thoroughly, constantly destroying old building blocks and creating new ones. But like single point crossover, double point crossover has also a conservative side and it is good at swapping entire genes. Consider Figure 4.3 (b) below. Similar to the previous example, individually both paths are not good however, when these two paths are crossed (twice this time), they produce two much better paths. Often times double point cross over will give much better results compared to single point cross over.



(a)



(b)

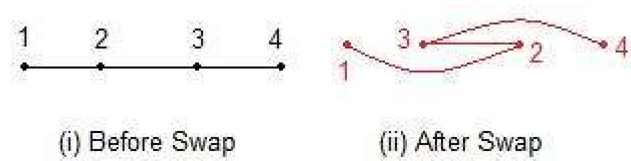
Figure 4.3: Double Point Cross Over

4.2.3 <u>Randomization</u>

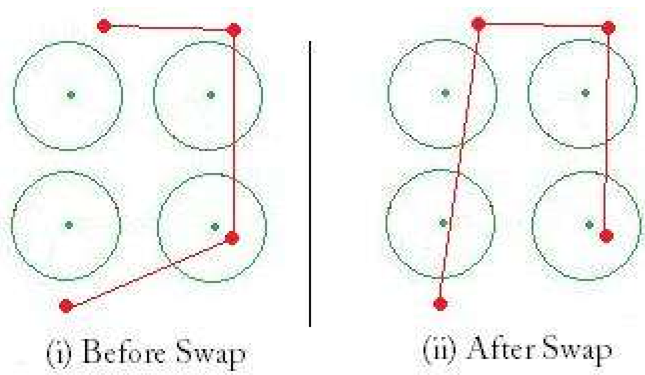
Randomization is an extreme mutation operator, where all geneses in the genetic code are mutated. Consider a case where the genetic algorithm is stuck in a local minimum (most of the parents are either mutations of the same path or offspring of these mutations). Changes made by mutation, crossover and many other operators may not be enough to break out of this local minimum. So a randomly created path that is inserted into the population may have a small chance to introduce new genetic material and thus rock the GA out of the local minimum/maximum.

4.2.4 Random Swap

Random swap is another version of mutation. Here instead of randomly mutating one element in the genetic code, we are swapping two (random) elements of it with each other. In our case this will result in reordering the points of the path. Figure 4.4 shows an example where if the mobile covers the same point but on a different path, it communicates with more sensors thus result in a more accurate localization.



(a)



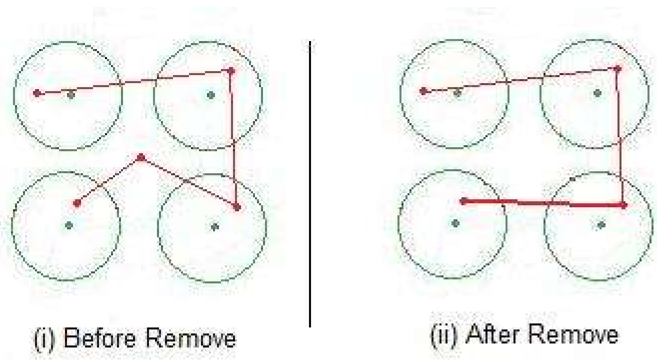
(b)

Figure 4.4: Random Swap

4.2.5 Random Remove

Removing a point from the path being traveled is another form of randomization. The primary goal of removing a point is not to increase the accuracy but to make the path shorter while also maintaining the overall localization accuracy. A point is randomly removed from one the paths and the path is evaluated to see if this has created a better path compared to the last path. Figure 4.5 shows an example of a path from which a point is removed at random.

(a)



(b)

Figure 4.5: Random remove

4.2.6 Greedy Path

Greedy path can be seen as genetic engineering. Greedy path is essentially an engineered reordering of the genetic material of an individual. In our case this will result in the same point set but a whole different trajectory over the point set. To understand why we have chosen greedy path, consider a grid in which sensors are deployed randomly. It is likely that a path will be created that will take the inefficient route and connect points on the extreme edges of the area. Even though this will result in a better overall localization, however, it is possible to shorten the path while maintaining the overall localization accuracy by connecting nearby points first and then connecting the furthest points.

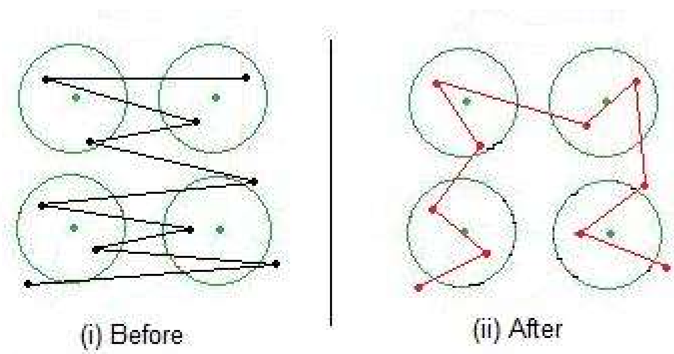


Figure 4.6: Greedy path

4.3 Terminating Condition

In theory GAs can run for unlimited period of time. However, cost of finding a better result increases every time a better result is found. Thus terminating conditions are used. Common terminating conditions include stopping after a fixed number of generations or finding a solution that satisfies minimum criteria or reaching a time/cost limit or a solution has been reached such that successive iterations no longer produce better or significantly better results or a combination of any of these conditions. In our experiments the process continues until the improvements with each generation become insignificant for a considerable period of time.

CHAPTER 5

IMPLEMENTATION

In this chapter, we describe the implementation details of our genetic algorithm approach to finding good paths for mobile beacons. In our setup, we consider sensor nodes that are uniformly deployed over a rectangular area. Each of the sensors are assumed to have the same communication distance. These sensors are not aware of their location. Only the mobile node is equipped with a GPS device.

5.1 Fitness Evaluation

Movement of the mobile node is defined as a straight walk from one point to another. These points are stops where the mobile node changes its direction. In other words the path of the mobile node comprises of a list of points. Mobile node then moves from one point to another in the order in which these points were assigned. It moves with a constant speed in the deployment area. Once a new path is created, its fitness level is evaluated and based on that fitness level it is decided if this path will be kept or discarded. After a new path is created, points where the mobile node will broadcast its location are calculated using the path, speed of the mobile node and broadcast interval. This interval is defined by the user in terms of distance. Once these points are calculated we cross reference these with the sensors location and calculate where the sensor received communication from. After the mobile node completes its walk, we calculate Cramer Rao bound for each sensor. Finally CRB values for all sensors are averaged which is the return value of the evaluation function. Once all paths for a generation are evaluated, they are sorted by their CRB value. Only the paths that have the lowest CRB value are kept for the next generation and rest are discarded.

5.2 Genetic Algorithm Implementation

Once selection of parents has happened, the parents will need to spawn offspring for new generations using mutation and crossover. There are many ways these operators can be implemented. In this thesis we have not only implemented the basic mutation and crossover operators, we have also implemented four modified operators. These operators are:

- Random
- Random Swap
- Random Remove
- Greedy Path

These operators are explained in detail in chapter 4 of this thesis. We randomly pick an operator using the assigned probabilities. Once the operator is decided, we pick a path from the parent to apply this operator. If the path that is created after applying this operator passes the distance constraint, we keep it, else it is discarded and we pick another operator and path on random. This process repeats until we have the required number of paths for each generation.

Each operator is assigned a probability. These probabilities play an important role in how good the solution is and how fast it can be found. Finding the right combination for these probabilities for each operator can be very challenging. Take mutation for example. If the chromosome is mutated, normally a few bits are changed and so the solution is a bit changed. But if all bits in chromosome are mutated, then the chromosome string is in fact random which is almost like no mutation. Probability of these operators also depends on the problem definition. Considering different problems, one set of probabilities may work well for one problem but may not be ideal for another problem. After running some experiments we found out that better result are achieved when probability of mutation is 0.15, randomization is 0.1, single point crossover is 0.2, double point crossover is 0.2, random remove is 0.1, random swap is 0.15 and probability for greedy path is 0.1.

CHAPTER 6

RESULTS

In this chapter, we discuss various experiments conducted and present their results. For our experiments we considered a grid of 500m by 500m with 100 sensors deployed uniformly in this grid. Unless specified by the user, each of these sensors has a range of 50m. We then generate 2500 paths in each generation and after evaluating keep only the best 100 of these paths. These paths also have limitation on the maximum distance covered. If no better path is found in 100 generations, the program terminates and saves the path with the lowest CRB. We performed several experiments in which we analyzed CRB with respect to maximum distance traveled by the mobile beacon and range of the sensors.

First we evaluate path created with each generation. Figure 6.1 shows how the best path changes with new generations. Initially the path created is not very good and has a very high CRB value. With every new generation as new paths are created, it covers more and more sensors and the CRB is improved. Initially the CRB value of the best path decreases very quickly however, as better paths are found, it becomes harder and harder to find an even better path and so the change in CRB values becomes little and little. We expect this value to be monotone decreasing with time, converging to the optimal solution. This can be seen in Figure 6.2.

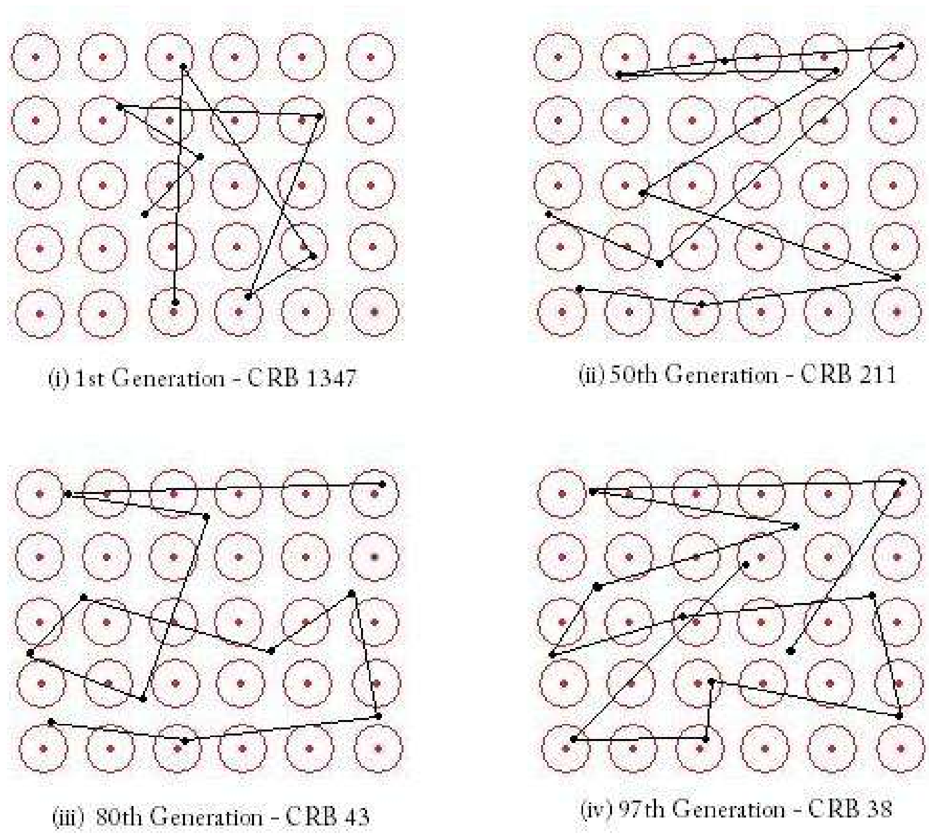


Figure 6.1: Path Representation and CRB

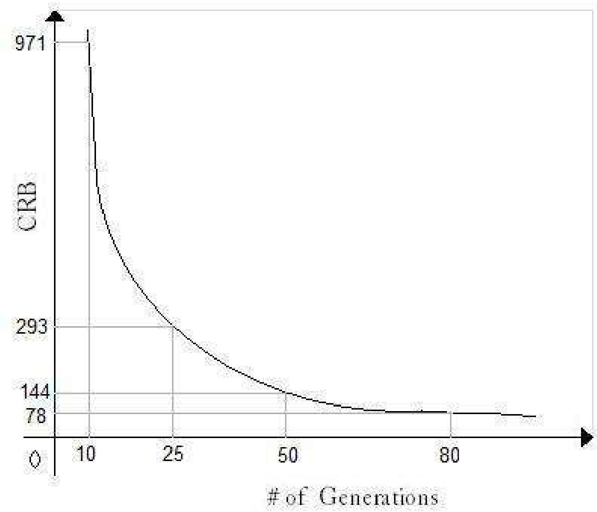


Figure 6.2: CRB vs. Number of Generations

To evaluate the effect of the maximum path distance on CRB we performed several experiments and varied maximum allowed distance from 2500 m to 20000 m. Figure 6.3 shows the results of this experiment. It can be seen that as we increase the maximum allowed distance of the mobile path, CRB decreases. In chapter 3 of this thesis we went over some factors that influence CRB directly. Path length was one these factors. Figure 6.3 supports this claim. A longer path means that sensors nodes can communicate with the mobile node more and thus resulting in comparatively more accurate localization.

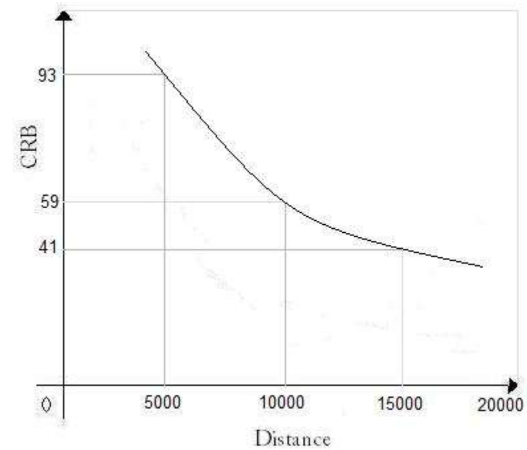


Figure 6.3: CRB vs. Maximum Distance

Another factor that influences CRB directly is the broadcast range of the sensor. A larger range translates in to more readings from the mobile node, which results in better localization and a lower CRB. For our experiments we decided to vary the range of the sensors and mobile node from 10 m to 80 m and fix the maximum path distance to 15000 m. The result of this experiment can be seen in figure 6.4.

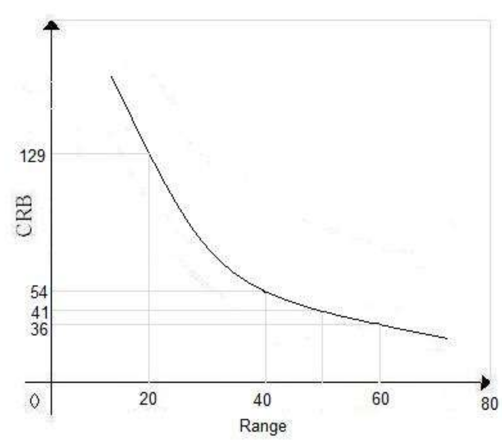


Figure 6.4: CRB vs. Sensor range

CHAPTER 7

CONCLUSION AND FUTURE WORK

This work described a genetic algorithm based approach to approximate an optimal path for a mobile beacon node in a grid of stationary wireless sensors. As the beacon moves over the field of sensors it broadcasts its location. Sensors that are currently in the proximity of the beacon will receive this communication and can then use several of these messages to compute estimates on their locations. An optimal path was defined as a path that will result in the highest overall precision of location estimates among sensors given a maximum path length for the beacon. We assumed that sensors are uniformly deployed in a predefined deployment area. We evaluated location precision calculating the maximum achievable accuracy using Cramer Rao Bound (CRB) for unbiased evaluation.

We described the path of the mobile beacon using strings of 'X' and 'Y' coordinate pairs. As paths are described using strings, they lend themselves to genetic algorithm manipulations. Thus, to improve on the localization precision given by the path of the mobile beacon, a genetic optimization approach is used. Multiple genetic operators including mutation, splicing, selection and cross-over were used to create new paths which were evaluated for precision. Details of the genetic optimization approach to find better and better generations of paths were given. Extensive optimization simulations were performed in order to look for paths resulting in high precision. We have shown the best paths found as well as looked at the relationship of maximum path length versus precision of overall location estimates.

Our solution focuses on a single mobile beacon node. However, localization could also be performed using multiple nodes. Additionally, if sensors were deployed in a non uniform manner over the target area (e.g. they could be concentrated around two more interesting points, or their distribution could follow a 2 dimensional Gaussian distribution) then mobile beacon paths would be significantly different. It will be interesting to use two mobile beacons, one for each area and minimize the total distance traveled by both beacons.

Also in this thesis we made an assumption on the target area for sensor deployment. We assumed it to be square for all our experiments. It will be interesting to see how this algorithm behaves in different shaped areas (e.g. circles and irregular shaped areas).

Earlier in this thesis we briefly touched on the problem of collinear readings from the mobile node. We also mentioned that to accurately localize a node path must be chosen in a way that each node receives at least three non-collinear readings from the mobile beacon. In this algorithm our mobile node moves from point A to point B in a straight line which makes it impossible to accurately localize in a single pass. It will be interesting to see the effect on distance traveled and localization error using the S-Curves presented in [5].

APPENDIX A

PSEUDO GENETIC ALGORITHM

```
initialize population (p) randomly
repeat until exit criterion is met
{
        determine fitness of each member in the population
        select the best n members of the population and define them as parents for the
                next population
        throw out remaining (p-n) members
        repeat until new population == p
        {
                select randomly from the next set of operators
                {
                        perform crossover on parents
                        perform mutation of parents
                }

                check if new child(ren) created by the previous operation satisfies the
                        constraints.
                if constraints are satisfied, accept child(ren) as a member in the new
                        population
        }

        check if exit criterion is met
}
```

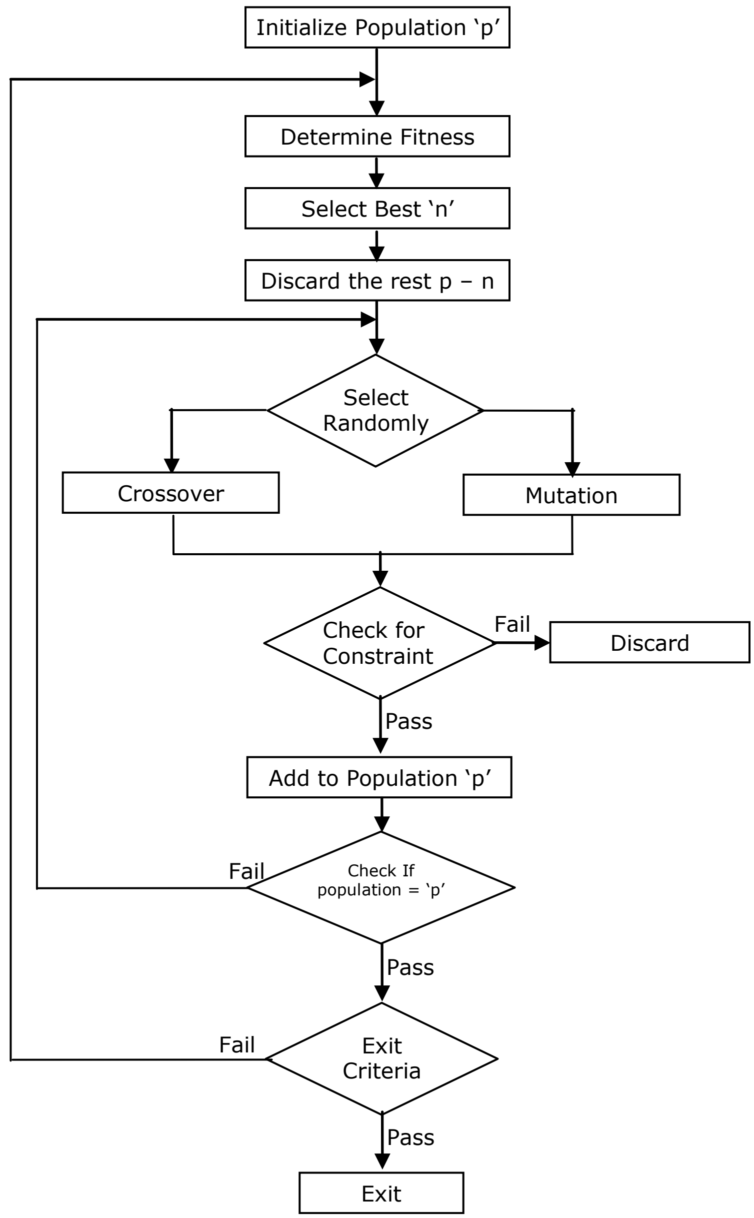


Figure A.1: Flowchart of genetic algorithm employed

REFERENCES

[1] A. Ward, A. Jones, and A. Hopper. “A new location technique for the active office”. In IEEE Personnel Communications, 1997.

[2] Chris Karlof and David Wagner. “Secure Routing in Sensor Networks: Attacks and Countermeasures”. First IEEE International Workshop on Sensor Network Protocols and Applications, May, 2003.

[3] D. Koutsonikolas, S. M. Das, and Y. C. Hu, “Path Planning of Mobile Landmarks for Localization inWireless Sensor Networks,” in Proc. of IEEE Distributed Computing Systems Workshops, pp. 86-86, 2006.

[4] Hocaoğlu, Cem & Sanderson, Arthur C. (1998). “Multi-dimensional path planning using evolutionary computation”. Proceedings of the IEEE Conference on Evolutionary Computation, 165-170.

[5] Huang and Zaruba. "Static Path Planning for Mobile Beacons to Localize Sensor Networks." percomw 00 (2007): 323-330.

[6] Ko, Young B. and Nitin H. Vaidya. "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks." Mobile Computing and Networking. 1998, 66-75.

[7] M. L. Sichitiu and V. Ramadurai, “Localization of Wireless Sensor Networks with a Mobile Beacon,” in Proc. of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004), pp. 174-183, Fort Lauderdale, FL, October 2004.

[8] N. J. Nilsson. “A mobile automaton: An application of artificial intelligence techniques”. In 1st International Conference on Artificial Intelligence, pages 509–520, 1969.

[9] N. Patwari, A. Hero, J. Ash, R. Moses, S. Kyperountas, and N. Correal, “Locating the Nodes: Cooperative Geolocation ofWireless Sensors,” IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 54-69, July 2005.

[10] S. Ratnasamy, D. Estrin, R. Govindan, B. Karp, S. Shenker, Yin, and F. Yu. “Data-centric storage in sensornets”. In Proceedings of the First ACM SIGCOMM Workshop on Hot Topics in Networks, Oct. 2002.

[11] Ssu, Kuo-Feng, Chia-Ho Ou, and H. C. Jiau. "Localization with mobile anchor points in wireless sensor networks." Vehicular Technology, IEEE Transactions on 54 (2005): 1187-1197.

[12] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher. “Range-free localization schemes for large scale sensor networks”. In Proceedings of the 9th annual international conference on Mobile computing and networking, pages 81–95. ACM Press, 2003.

[13] Yih-Chun Hu, Adrian Perrig and David Johnson. “Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks”. IEEE InfoCom 2003. April 2003.

BIOGRAPHICAL INFORMATION

Awais Iqbal received his Bachelors degree in Computer Science from Franklin University, Columbus Ohio in December 2005. He completed his early education in Pakistan and then moved to the USA in late 2002 to purse his Bachelors degree. Before starting his graduate studies he worked for an IT company in Columbus Ohio. In December 2006, he moved to Texas and joined University of Texas at Arlington to purse a Master of Science degree in Computer Science. His research interests include wireless networking, localization and geographic information systems (GIS).