2007

Improving the lifetime of two-tiered sensor networks using genetic algorithm

Md. Shamsul Wazed
University of Windsor

Recommended Citation
Wazed, Md. Shamsul, "Improving the lifetime of two-tiered sensor networks using genetic algorithm" (2007). *Electronic Theses and Dissertations*. 4662.
https://scholar.uwindsor.ca/etd/4662

Improving the Lifetime of Two-Tiered Sensor Networks Using Genetic Algorithm

By
Md Shamsul Wazed

A Thesis
Submitted to the Faculty of Graduate Studies
through the School of Computer Science
in Partial Fulfillment of the Requirements for
the Degree of Master of Science at the
University of Windsor

Windsor, Ontario, Canada
2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-34920-5
Our file Notre référence
ISBN: 978-0-494-34920-5

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Abstract

Wireless sensor networks have been envisioned to have a wide range of applications which consist of many inexpensive and low-powered wireless nodes which are used to sense, gather, and transmit the data towards the base station. In Two-Tiered wireless sensor networks, nodes are grouped into clusters, with a minimum of one cluster-head to distribute the work load among the member nodes. In the recent years, higher-powered relay nodes have been proposed to act as cluster heads and these relay nodes form a network among themselves in order to improve the lifetime of the sensor networks. Since the nodes are generally energy constrained, efficient management of the network data communication scheme can maximize the lifetime of the networks.

A Genetic Algorithm is the technique for randomized search and optimization which is based on Darwin's Principal of Natural Selection. In this paper, we have proposed a Genetic Algorithm based solution for scheduling the data gathering of relay nodes that can significantly extend the lifetime of the relay node network. We have simulated our method on 15 different sizes of networks and measured the lifetime of the network as the number of rounds, until the first relay node runs out of battery power. For smaller networks, where the global optimum can be determined, our genetic algorithm based approach is always able to find the optimal solution with a lesser program run-time. For larger networks, we have compared our approach with traditional routing schemes and shown that our method leads to significant improvements.

Dedication

This work is dedicated to my late father Mr. Abdul Wahab, who died 6 years ago and my mother, who lives far away in my home country, Bangladesh.

Acknowledgements

I would like to acknowledge my sincere gratitude towards my supervisor Dr. Arunita Jaekel for her valuable advice and suggestions, without which this work would never been be achieved. I would also like to thank Dr. Subir Bandyopadhaya for his guidance during the development of this thesis. Special thanks to Ataul Bari for his cooperation and encouragement to me during my study period. Finally, I would like to thank my wife Mrs. Mahamuda Khatun for her support, and patience to complete my master degree.

Table of Contents

CHAPTER

List of Figures

List of Tables

List of Abbreviations

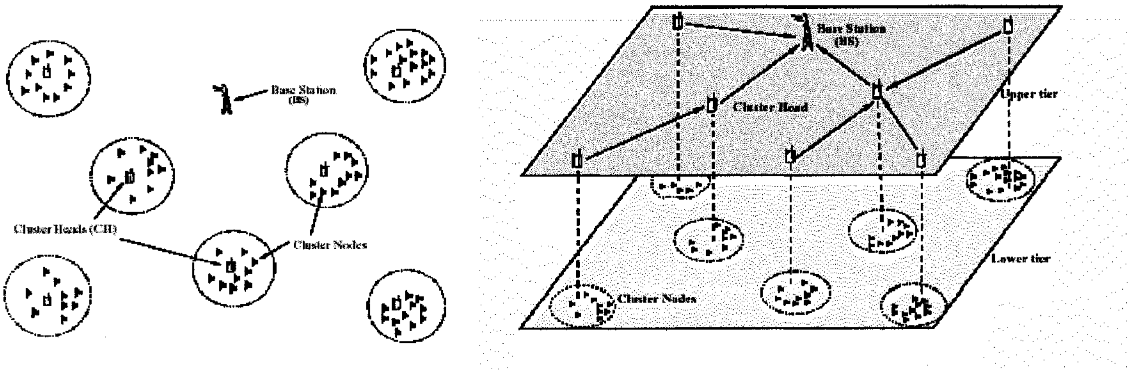
WSN	:	Wireless sensor network
GA	:	Genetic Algorithm
MHDTM	:	Multi-Hop Data Transmission Model
MHRM	:	Minimum Hop Routing Model
MTEM	:	Minimum Transmission Energy Model
ILP	:	Integer Linear Programming
UDRR	:	Uniform Data Receiving Rate
NDRR	:	Non-uniform Data receiving Rate
CI	:	Confidence Interval
BS	:	Base Station

CHAPTER 1

Introduction

A Wireless Sensor Network (WSN) consists of a collection of inexpensive, lightweight, battery-operated multifunctional sensor nodes. The feasibility of these sensor networks has been accelerated through the advances in MEMS (Micro Electro-Mechanical Systems) technology, combined with low power, low cost Digital Signal Processors (DSPs), and Radio Frequency (Rf) circuits. Sensors nodes are often severely energy constrained and expected to last until their energy drains out. Since it is not practical to replace or recharge the batteries of thousands of sensor nodes deployed in hostile territories, the key challenge in sensor networks becomes maximizing the lifetime of the sensor nodes.

To balance the network transmission load among the nodes, WSNs are sometimes grouped into clusters. These cluster-based sensor networks can implement a two-tiered architecture. In the lower tier, the sensor nodes are deployed in each cluster to sense and transmit data to the cluster-head. There is one cluster-head in each cluster (Fig 1.1). The cluster-head is responsible for receiving raw data from the sensor nodes, creating a local-view by exploring application-specific correlations among the data, and finally, forwarding data towards the base station, either directly or via other cluster-heads. These cluster-heads and the base-station constitute the upper tier of two tiered WSNs [Pan05].



(a) Physical topology (b) Logical topology

Figure 1.1 : Reference architecture for a two-tier WSN [Hou05]

1.1 Problem Outline

Sensor networks consist of a large number of battery powered sensor devices. Different cluster heads or relay nodes normally drain out their batteries at different times, resulting in parts of the network becoming inactive. The lifetime of a WSN can thus be defined as the time from the inception of the sensing operation of the network, till a fraction of the nodes deplete their power. The node which runs out of its battery power first, is known as *critical node*.

To maximize the lifetime of the network is to extend the lifetime the of critical node(s). To reduce the burden on the sensor nodes, some special functionality nodes are deployed in hierarchical WSN, which are known as *relay* nodes [Bari06a]. These relay nodes act as cluster-heads, and can be provisioned with higher energy compared to the sensor nodes. These relay nodes form a network among themselves, and transmit data towards the base station using multi-hop paths. The relay nodes can also be used to improve connectivity and better load balancing for the networks. Similar to sensor nodes, relay nodes are also battery-operated devices capable of wireless communication and hence, power constrained. Thus, to extend the lifetime of the network, it is necessary to find out the optimal data gathering and routing scheme.

1.2 Existing Solutions

In hierarchical sensor networks, higher-powered relay-nodes have been used as cluster heads in the last few years. These relay nodes route data towards the base station of the network. In such networks, maximizing the lifetime of two-tiered network directly depends on maximizing the lifetime of the network relay nodes.

Most papers deal with the data routing of relay nodes according to a *flow-splitting* [Hou04, Bor05, Yar05, Bari06] model. In a flow-splitting model, data transmits from a single node towards multiple destination nodes simultaneously. This approach has a number of limitations, including the requirement by the relay nodes to perform complex routing functions and costly packet level power control for nodes that are equipped with a single transmitter [Hou04].

In the paper of Bari et. al., the authors proposed an ILP formulation to find out the optimal routing scheme for a non-flow splitting relay-node network to extend the network lifetime. The authors compared their formulation with a direct energy transmission model to show significant improvements. But these formulations are not suitable for large sensor networks and computationally intractable for even a moderate sized network. For example, the CPLEX [ILCP] runtime of these formulations takes more than 60 hours to optimize the routing of a network of size 22 (one base station, with 21 relay nodes which act as cluster heads) with the machine of 2.4 GHz processor and 512 MB RAM under UNIX environment. CPLEX was unable to produce any solution for networks of size larger than 25 within a reasonable amount of time.

1.3 Proposed Approach

In this thesis, we propose a Genetic Algorithm (GA) based solution for scheduling the data gathering of the relay nodes that improves the lifetime of the network. The relay nodes act as cluster heads to collect data from their respective clusters, or from neighboring relay nodes, and forward this data towards the base station either directly or via other relay nodes using a *multi-hop data transmission* (MHDT). Each period of data gathering and transmitting to the base station is referred as a *round* [Bari06a, Bari06b, Kal02] and the lifetime of the network is represented in terms of the number of rounds until the first relay node runs out of its battery power. The *non flow-splitting* data routing is considered assuming both fixed and variable amounts of data received by each relay node from the sensor nodes of its own cluster. We assumed that the nodes are stationary after deployment. The number of relay nodes, and their positions, are also assumed to be known. The initial energy of each relay node is considered to be equal. The transmission ranges of each sensor and relay node are considered as mentioned in [Tan06].

1.4 Why Genetic Algorithm ?

The GA approach can be used to solve optimization problems based on the genetic processes of biological organisms. It is useful to find out the approximate solutions to difficult-to-solve problems, like NP-Complete problems. The major advantage of genetic algorithms are their flexibility and robustness as a global search method. GAs are not guaranteed to find the global optimum solution to a problem, but they are generally good at finding *acceptably good* solutions to the problems in a reasonable amount of time. In this thesis, we consider large networks and determine a routing scheme that extends the lifetime of two-tiered WSNs. The Optimal Integer Linear Program (ILP) formulations can be used effectively only for small networks. Even those take higher computational time than GA. Special care is needed while designing some GA parameters like population size, rate of cross-over, rate of mutation etc. to achieve more suitable solution.

1.5 Thesis Organization

This thesis is organized as follows. Chapter 2 reviews some relevant background information. Chapter 3 describes the network model used in this thesis. Chapter 4 describes the methodology used to generate the population and implement the genetic algorithm. The experimental results, comparisons and discussions, are presented in Chapter 5. Chapter 6 summarizes our conclusions and points to directions for further work.

CHAPTER 2

Background Information

2.1 Two-Tiered Wireless Sensor Networks

Over the last few years, the design of sensor networks has gained increasing importance due to their many potential applications. Such networks are expected to play a significant role in a wide range of applications, including motion detection, environment monitoring, military surveillance, and reconnaissance. Maximizing the network lifetime is a key challenge in most of the research because the nodes are severely energy constrained. Most of sensor nodes are equipped with a small battery that can store, at most, 1 Joule [Kah99]. Also, it is not feasible to replace or recharge the batteries of thousands of deployed sensors. Therefore, the battery charge taken with them to the field must be conserved in order to extend the life time of the individual sensor nodes, as well as the entire sensor network. To maximize the lifetime of two-tiered sensor networks, researchers have proposed several aspects which include :

- Clustering algorithms
- Data gathering and routing schemes
- Task allocation algorithm for nodes

Sensors in a WSN are often grouped into clusters, each with its cluster head. Instead of communicating directly with the sink (base station), a sensor transmits data to its cluster head, and each cluster head relays them to a sink node directly, or through other (intermediate) cluster heads. Thus, a cluster head closer to the sink needs to relay more traffic towards the sink, which can quickly consume the nodes' batteries. In case of the early *death* of any cluster head, the underlying sensors covered by the failed node (cluster head) will become inaccessible (although they may be fully functional) and make the network lifetime shorter.

2.1.1 Applications

Recent advances in wireless communications and electronics have enabled the development of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate over a short distances. Wireless sensor networks can be deployed both indoors and outdoors, substituting for our sensory organs in inaccessible or inhospitable areas. For different application areas, there are different technical issues that researchers are currently resolving. Depending on the deployment platform, there are a variety of applications for such sensor networks. A sensor can be used to measure humidity, temperature, illumination, and pressure; detect movement, including speed and direction; habitat monitoring; or to detect an object's shape and size. Sensor networks are widely used in a variety of areas, including military, security, health, and environmental applications. In military applications, sensor networks are used for target detection, to monitor forces, equipment and ammunition, surveillance of battlefield, and to detect nuclear, biological, or chemical attacks. For environmental applications, sensors can be used for forest fire detection, flood detection, detecting environmental pollution, agricultural precision, etc. It can be also used in seismic activity monitoring. Health applications include patient-condition monitoring, diagnostics, and hospital-drug administration.

2.1.2 Types of Nodes

Typically, a WSN is composed of a large number of nodes with processing, sensing and radio communication capabilities. In cluster-based sensor networks, each cluster has exactly one cluster-head, which acts as a cluster manager. Each cluster head collects data from the sensors in its cluster and relays them to a sink node directly or through other cluster heads in the upper tiers [Shu05]. These cluster heads are sometimes called *gateway* nodes [Yon03] or *application* nodes [Pan05]. To reduce the burden on these overloaded nodes, some special nodes are deployed in WSN, which are known as *relay* nodes [Bari06a, Bari06b]. Similar to sensor nodes, relay nodes are also battery-operated devices capable of wireless communication, but their job is to relay data generated by other sensor nodes without sensing the environment.

2.1.3 Node Sub-systems

A sensor node usually consists of four sub-systems [Rag02], as described below :

- A computing subsystem - It consists of a microprocessor (microcontroller unit, MCU), which is responsible for the control of the sensors and the execution of communication protocols. MCU's usually operate under various operating modes for power management purposes.
- A communication subsystem - It consists of a short range radio, which is used to communicate with neighboring nodes and the outside world. Radios can operate under the Transmit, Receive, Idle, and Sleep modes. It is important to completely shut down the radio rather than put it in the Idle mode when it is not transmitting or receiving because of the high power consumed by this mode.

- A sensing subsystem - It consists of a group of sensors and actuators that link the node to the outside world. Energy consumption can be reduced by using low power components and saving power at the cost of performance, which is not required.
- A power supply subsystem - It consists of a battery which supplies power to the node. The lifetime of a battery can be increased by reducing the current drastically or even turning it off often.

2.1.4 Types of WSN

The WSNs structure can be divided as follows according to the routing protocols [AlK05] :

- *Flat Network Routing* – In this network, each node typically plays the same role, and all sensor nodes work together to sense the data. Due to the large number of sensor nodes in this structure, it is become feasible to assign a global identifier to each node. The base station sends queries to certain regions and waits for data transmissions from the sensor nodes located within that region.

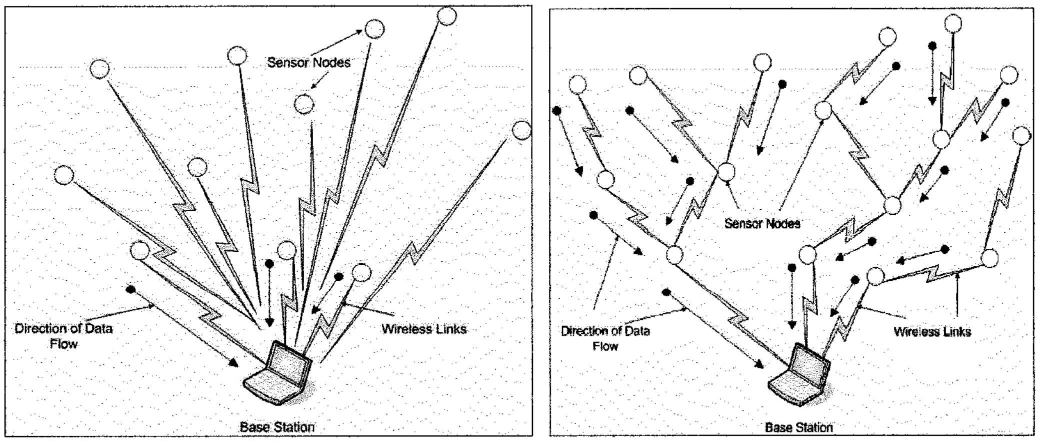


Figure 2.1 : Single-Hop and Multi-Hop routing for Flat Network

- *Hierarchical Routing* – In hierarchical or cluster-based architectures, higher energy nodes (cluster-heads) are responsible for collecting data from their respective cluster and forwarding them to the base station. To perform the sensing, low energy sensor nodes are used. This is a well-known technique with special advantage of network scalability and efficient communication. LEACH, APTEEN (TEEN), and PEGASIS are some example of hierarchical clustering protocols. In hierarchical routing, data communication from relay nodes to the base station may be either *single-hop* or *multi-hop*. In the single-hop data transmission model [Hei00, HTh00], the cluster heads send data directly to the base station, provided that the base station lies within the transmission range of all the relay nodes. In the multi-hop data transmission model [Kal02, Hou04, Hou05, Tan06], relay nodes not only transmit data that they receive from the sensor nodes belong to their respective clusters, but also forward data from other relay nodes towards the base station.
- *Location-based Routing* – In this kind of routing in WSN, sensor nodes are addressed by means of their corresponding locations and the distance between the neighboring nodes can be measured on the basis of incoming signal strength.

Cluster-based sensor network can also be classified into two broad types – *homogeneous* and *heterogeneous* sensor networks. In homogeneous networks all the sensor nodes are identical in terms of battery energy and hardware complexity. The cluster-head nodes transmit data towards the remote base station, and are also responsible for data aggregation and protocol co-ordination. Heterogeneous sensor networks contain two or more different types of nodes with different battery energies and functionality. It can reduce the network cost by embedding a smaller number of cluster head nodes. As cluster heads are fixed in this network, the rotation of cluster heads (for load balancing purposes) is no longer possible [Mha04].

Clustered sensor networks can also be again classified as *single hop* and *multi-hop* network. A single hop network is one in which sensor nodes transmit the data directly to the cluster head without relaying it through any intermediate node. In a multi-hop network, nodes use multi-hopping to reach the cluster head by using intermediate node, or nodes. This is particularly the case when the propagation loss index for in-cluster communication is large ($m > 2$) since the sensor nodes are deployed in regions of dense vegetation or uneven terrain. In such cases, it may be beneficial to use multi-hop communication among the nodes in the cluster to reach the cluster head [Mha04].

Besides the two-tier WSN as described above, a Three-Tier WSN architecture has also been proposed by Zha et.al. which is designed specifically for the traffic monitoring and application processing purposes. At the *Sensing & Fusing Tier*, the networked sensors in clusters monitor the physical traffic and transmit data to their local cluster head. The *Transmission & Management Tier* equipped with high powered cluster heads to provide reliable, long-range, and high data rate transmission among communications at this tier. The proposed last tier, the *Decision-making Tier*, is claimed to provide transportation statistics and traffic control assistance service strategies owing to aware the information supplied by the other two tiers.

2.1.5 Limited Resources

A wireless sensor network is a special network which has many constraints compared to other networks. One of the basic characteristics of sensor networks is that they are expected to perform in an unattended manner, i.e. without human intervention. Sensor nodes are also powered by low-powered batteries. The functions of sensing and communicating require a certain amount of resources for implementation, including data memory, code space, and energy to power the sensor. However, currently these resources are very limited in a tiny wireless sensor. The sensor nodes have the following resource constraints:

- Limited Memory and Computational power – A sensor is a tiny device with only a small amount of memory and storage space for the code. Its' computing power is also limited. For example, a MICA mote has an 8-bit, 4MHz CPU only with only 512 KB storage capacity (Table 2.1). This restricts the types of data processing algorithms on a sensor node, as well as the sizes of intermediate results that can be stored on the sensor nodes.

Processor	4Mhz, 8bit MCU (ATMEL)
Storage	512KB
Radio	916Mhz Radio (RF Monolithic)
Communication Range	100 ft
Data Rate	40 Kbits/sec
Transmit Current	12 mA
Receive Current	1.8 mA
Sleep Current	5 uA

Table 2.1 : Hardware Characteristics of a MICA Mote [Yao02]

- Limited Communication – The sensor nodes have latency with high variance and limited bandwidth. It has a 916 MHz radio transceiver that collets information from cluster members with communication rage of 100 ft (40 m) only.
- Limited Power – Limited power is the biggest constraint of wireless sensor networks. To keep the cost and size of the sensor nodes small, they are equipped with low powered batteries. For example, the MICA motes are powered by two AA batteries, that provide about 2000mAh [Hil02], powering the mote for approximately one year in the idle state and for one week under full load. Also, it is not feasible to replace or recharge the thousand of high cost of sensors. Therefore, the battery charge taken with them to the field must be conserved to extend the life time of the individual sensor node as well as the entire sensor network. The processing of security related functions (e.g., encryption, decryption, signing data, verifying signatures) consumes extra battery power.

2.1.6 Design Factors

WSNs have several restrictions which include limited non-replenishable energy supply, limited computing power, limited bandwidth of wireless links and small size of sensor nodes. One of the main design targets for WSNs is to carry out data communication while trying to maximize the lifetime of the network. Thus WSNs pose a number of unique technical challenges due to the following factors [AlK05 and Aky02] :

- *Energy consumption without losing accuracy* – Sensor nodes need to use their power efficiently due to the limited supply of energy as the included batteries are not rechargeable or replaceable. Proper power conservation and power management are needed to ensure the receiving of all data from source node to destination node without any packet lost or data conflict. Early power dissipation by any node causes significant topology changes and reorganization of the network routing.

- *Node Heterogeneity* – Depending on the application, a sensor node can have different roles or capabilities (heterogeneity) which raises many technical issues related to data transmission.
- *Unattended Operation and no Human Intervention* - As the network may be deployed in inhospitable territories, or as the number of nodes deployed in sensor networks can be very large, it may be physically or economically infeasible for any kind of human intervention after the deployment of the networks.
- *Fault Tolerance* – The failure of sensor nodes, due to a lack of power, physical damage, or environmental interface, are more frequent in sensor networks. The network should sustain information dissemination in spite of failures.
- *Scalability* – Network protocols should be able to scale to a high degree and take advantage of high density of networks with hundred or thousands of nodes.
- *Connectivity* – The sensor node should be highly connected with each other, and no node should be completely isolated from the other nodes.
- *Coverage* – Each sensor node obtains a certain view of the physical area which is limited both in range and in accuracy.
- *Data Aggregation* – Sensor nodes may generate significant redundant data, similar packets from multiple nodes. To reduce the number of re-transmission, data aggregation should be performed to send a single packet.

- *Quality of Service* – As conservation of energy is directly related to the lifetime of the network, there may be a trade-off between the quality of the result and conservation of energy, especially where maximizing the lifetime is more important than the quality of data sent. Therefore, energy-efficient schemes are needed to address the issue.

2.1.7 Sensor Networks vs. Ad-hoc Wireless Networks

Wireless sensor networks may appear to be similar to ad-hoc wireless networks, but several important distinctions can be drawn between the two.

- Ad-hoc networks typically support routing between any pair of nodes, whereas sensor networks have a more specialized communication pattern, like many-to-one, one-to-many, and local communication.
- In most of the sensor networks, nodes are not mobile, possibly embedded in walls or dispersed from an airplane in a field.
- Sensor networks are more resource constrained than ad-hoc networks. Nodes in an ad-hoc network may have a 32-bit processor, megabytes of RAM, a 2 Mbps radio, and a large battery, whereas a typical sensor node have an 8-bit processor, kilobytes of RAM, a 40 Kbps radio, and a tiny battery.
- There may exists a significant amount of redundancy in sensor network traffic as an event in the environment may cause several neighboring nodes to send data to the sink at correlated times. This redundancy is almost absent in case of ad-hoc networks.

2.1.8 Hierarchical Protocols

In a hierarchical architecture, higher energy cluster heads are used to perform data aggregation and fusion, and process to send the information to the base station. On the other hand, the low energy sensor nodes are only used in sensing the target. Thus the main aim of hierarchical routing is to efficiently maintain the energy consumption of the sensor nodes and to maximize the network lifetime. Some initial and popular hierarchical routing protocols are described below –

- *LEACH (Low-Energy Adaptive Clustering Hierarchy)-* It is one of the first hierarchical routing schemes for sensor networks [Hei02]. The basic idea of LEACH is to organize sensor nodes into clusters based on the received signal strength. The node which requires the minimum communication energy for routing data to the base station is selected as cluster-head. The cluster-heads position change randomly over time in order to balance the energy dissipation of the nodes.
- *PEGASIS (Power-Efficient Gathering in Sensor Information Systems)* – This is a chain-based protocol [Lin02]. In PEGASIS, each node transmits data only to its close neighbor. The node which receives data from its neighbor first aggregates with its own data and then sends the data to the next neighbor on the chain. Unlike LEACH, PEGASIS uses only one selected node on the chain to transmit data to the base station instead of using multiple cluster-heads.
- *APTEEN (Adaptive Periodic Threshold-sensitive Energy Efficient sensor Network)* — APTEEN [Man02] is the *reactive* approach in LEACH to further enhance the energy efficiency in sensor networks. APTEEN combines both

proactive and reactive policies which allow the user to set threshold values and a count time intervals. The count time interval is the maximum time period between two successive reports sent by a node. If a node does not send data for a time period equal to the count time, it is forced to sense and retransmit the data.

Two widely used routing strategies, under the non-flow-splitting model are, the Minimum Transmission Energy Model (MTEM) and the Minimum Hop Routing Model (MHRM). In MTEM, each node *i* transmits to its nearest neighbor *j*, where *j* is closer to the base station than *i*. If there is more than one such node, only one is selected. In MHRM, each relay node transmits its data towards the base station as far as it can but within its transmission range maintaining the minimum numbers of hops to reach the base station. If the base station is within the transmission range of any source node, then data transmits directly to the base station. Fig 2.2 shows the sample of data routing for MTEM and MHRM respectively.

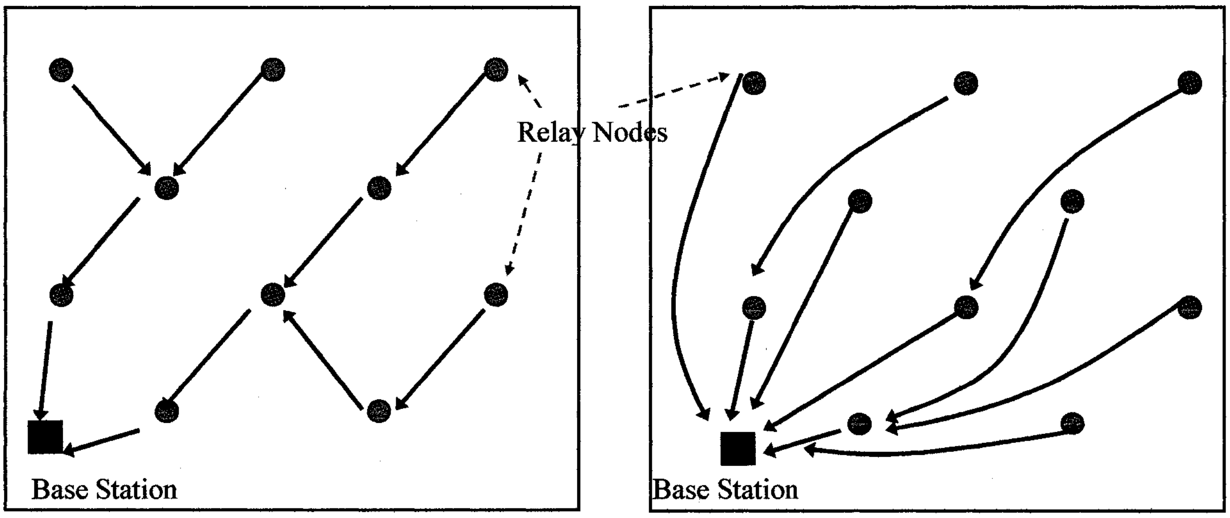


Figure 2.2 : Minimum Transmission Energy Model (MTEM) and
Minimum Hop Routing Model (MHRM)

2.2 Overview of Genetic Algorithms

Genetic Algorithms (GA) are techniques for randomized search and optimization, which is based on Darwin's Principal of Natural Selection [Gol89, Hol75, Rec73]. The most basic concept of the optimization is "Survival of the fittest", that is the strong will survive and the weak will die out. GAs employ a fitness function to determine how good the solution is for a particular given problem. The individuals having a greater fitness are chosen to reproduce the next generation. This process is repeated until an acceptable solution to the problem is found. Since GA is a heuristic, it estimates a solution to produce near to optimal results in a reasonable time.

The idea of evolution strategies in solving problems was first introduced by I. Rechenberg in 1973 [Rec73]. Later J. Holland [Hol75] and his students developed the concept of the genetic algorithm in 1975. Both of them developed a unique notation by using a mixture of biological and computer science term. GAs are now successfully applied in a wide range of studies in solving optimization problems, specially problems that are not well-structured and interact with a large number of possible solutions.

GA starts with a set of the randomly generated possible solutions (each of which is known as *Chromosome* or *Individual*) and the set is called a *Population.* Each chromosome is a simple string or array of *Genes* which contains a part of solution and the values of genes are called *Alleles.* The length of each chromosome should be same in a produced population. A *Fitness Function* is provided to assign the *Fitness Value* for the each individual. The function is based on how far or close an individual is from the optimal solution - greater the fitness value better the solution the individual contains.

Two randomly selected chromosomes, known as *Parents*, are twisted together in a process called *Recombination*, or *Crossover*, to produce two new chromosomes known as the *Child* or *Offspring*. If both the parents share a particular pattern in their chromosome, then the same pattern will be carried over to the offspring. To obtain a good solution, *Mutation* is often applied on randomly chosen chromosomes after the process of crossover. Mutation helps to restore any lost genetic values when the population *converges* too fast. Once the process of crossover and mutation occurs in a population, then the chromosomes for the *next generation* must be selected. To ensure that the new generation is at least as fit as the previous generation, some of the poorest performing individuals of the current generation can be replaced by appending same number of best performing individuals from previous generations. This is called *Elitism*. This cycle is repeated until the *stopping criterion* of the algorithm is met. The flow diagram of a typical GA process is shown in Fig 2.3.

The important factors that affect the GA performance are – Size of Population, Fitness Function, Method of Crossover, Mutation Probability, and Method of Replacement. The details of individual steps of a typical genetic algorithm are as follows :

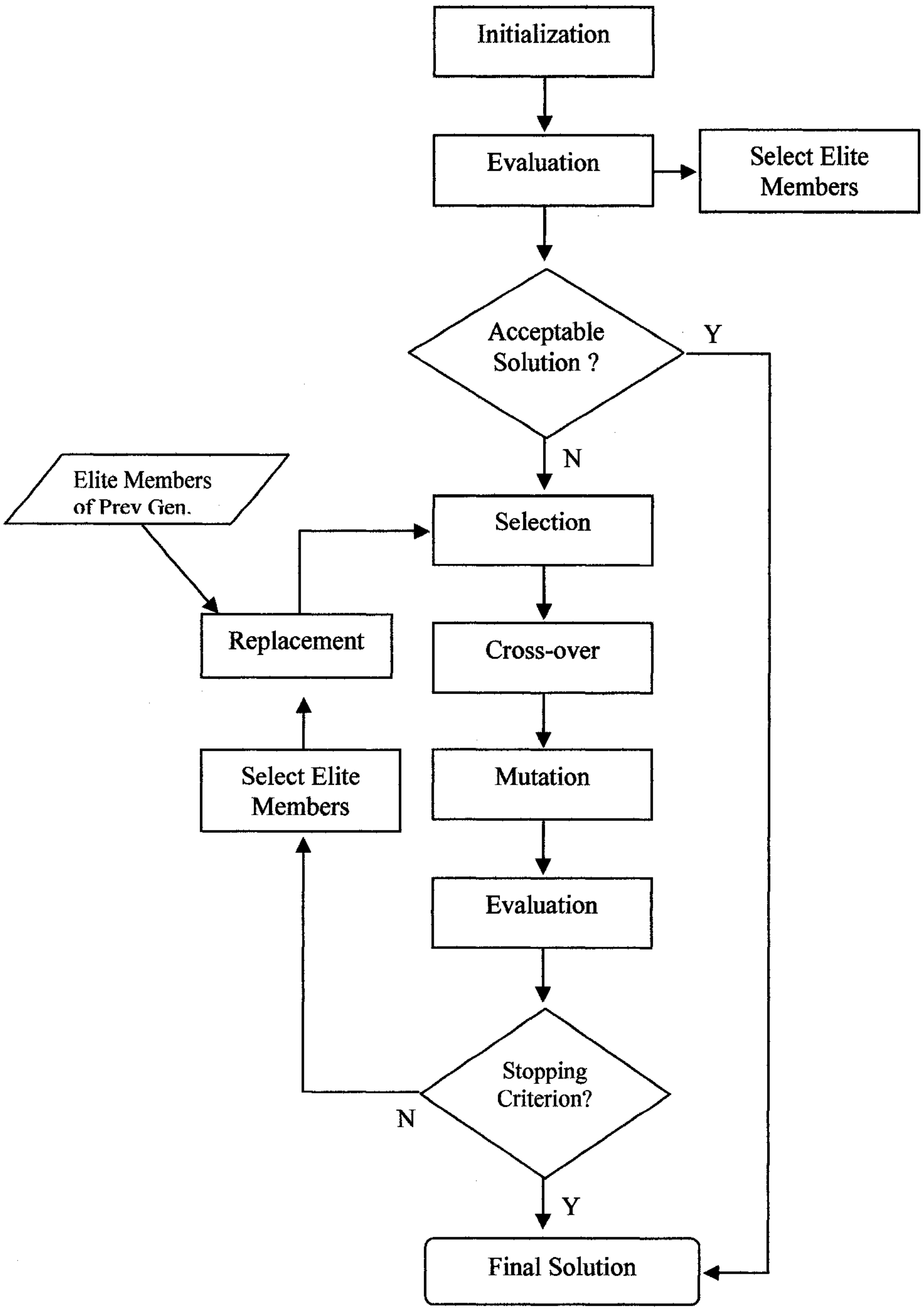


Figure 2.3 : Flow diagram of a GA process with Elitism

2.2.1 Initialization

The initial population (i.e. a set of chromosomes) is usually generated randomly. Each individual/chromosome of the population is a possible solution of the given problem. The individuals are generated satisfying the all criterion of the network. The most popular way to represent an individual is as a sequence of 0's and 1's. However, genetic algorithms can be applied with individuals that are the strings of integers, floating point numbers, or arbitrary numbers. Before generating the population, it is crucial to determine the size of the network. The size of the network greatly affects the problem solution. A small population size may lead to premature convergence before reaching an acceptable solution. On the other hand, if the population size is too large, it leads to unnecessary computational time [Sas05]. However, the size of the population should remain the same for all generations.

2.2.2 Evaluation

Each individual of the population is needed to determine its corresponding fitness value by using the fitness function. This function represents how close an individual is to the optimal solution. The individual with a fitness value represents a better solution to the problem. The most difficult and important step to implement the genetic algorithm is to construct the fitness function. The fitness function must also be fast because the fitness value of each individual must be calculated. Hence, the program runtime also depends on the fitness function of the genetic algorithm.

2.2.3 Selection

Selection is a process in GA to select the individuals/chromosomes with higher fitness values for the recombination (i.e. crossover) process. Individuals with a higher fitness value have a higher probability of being selected. Thus, GA imposes the *survival-of-the-fittest* mechanism. There are many methods for selecting the best chromosomes : Roulette-Wheel Selection [Hol75, Gol89, Dav91], Tournament Selection [Bri81, Gol91], Rank Selection, Steady State Selection etc, some of them are described below -

2.2.3.1 Roulette-Wheel Selection : This selection is based on *Fitness Proportionate Selection* method [Sas05]. Parents having better fitness values have a higher probability of being selected. In this method, each individual is assigned to the part of roulette wheel (Fig. 2.4) with the corresponding slot sized according to fitness. The wheel is spun *n* times to select *n* individuals. It is not guaranteed that the fittest individual will always to be selected. Similarly there is no guarantee for

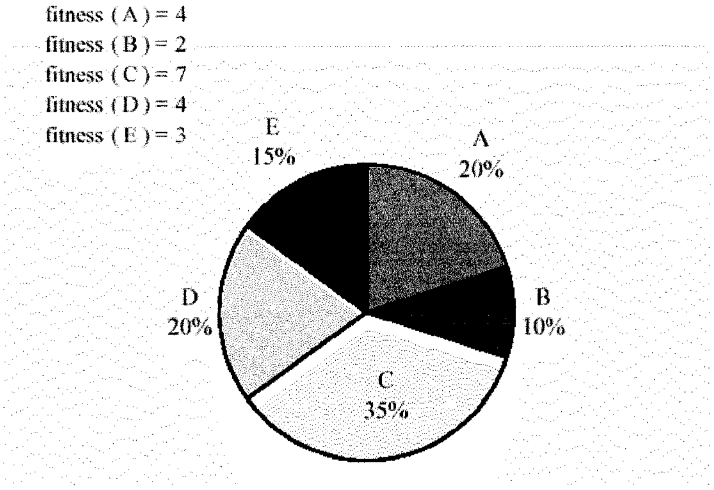


Figure 2.4 : Roulette-Wheel Selection

the lowest individual not to be selected. Again, there is a chance of selecting the same individual more than one time. Implementation of the roulette-wheel selection is briefly described in the section 4.4.

2.2.3.2 Tournament Selection : In Tournament selection process, *s* individuals are chosen randomly and entered into a tournament against each other. The individual with highest fitness value wins the current tournament and is selected as the parent. The selected parent is then just copied into the mating pool and then all tournament members (including selected member) are replaced into the original population. This is repeated until the mating pool is full. The most common type of tournament selection is *binary tournament*, where the value of *s* is 2. Using these selection processes, *n* tournaments are needed to select *n* individuals. Larger tournaments may also be used for selection [Bea93], from where the best of *m* (and $m > 1$) individuals are copied into the mating pool. This selection process increases the selection pressure, as below average individuals are less likely to win a tournament, while above average individuals are more likely to.

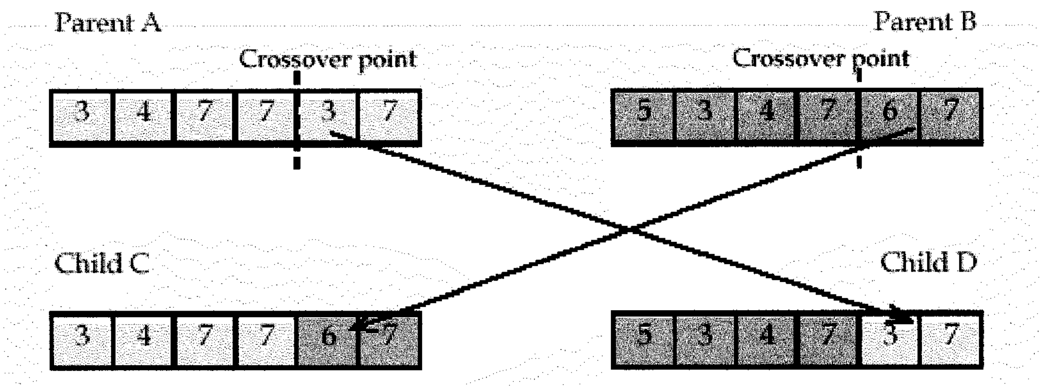
2.2.3.3 Rank Selection : In rank selection, individuals are sorted in order of raw fitness and assigned fitness from this ranking. The worst will have fitness 1, second worst 2, and so on, and the best will have the fitness *n*, where *n* is the size of population. This method is mainly used when the fitness differs greatly.

2.2.3.4 Steady State Selection : In a steady-state selection, one member of the population is changed at a time. To perform selection, each member of the population is chosen according to its fitness. The first selected member is copied and the copy mutated. The second member of the population is selected which is then replaced by the previously mutated member and so on.

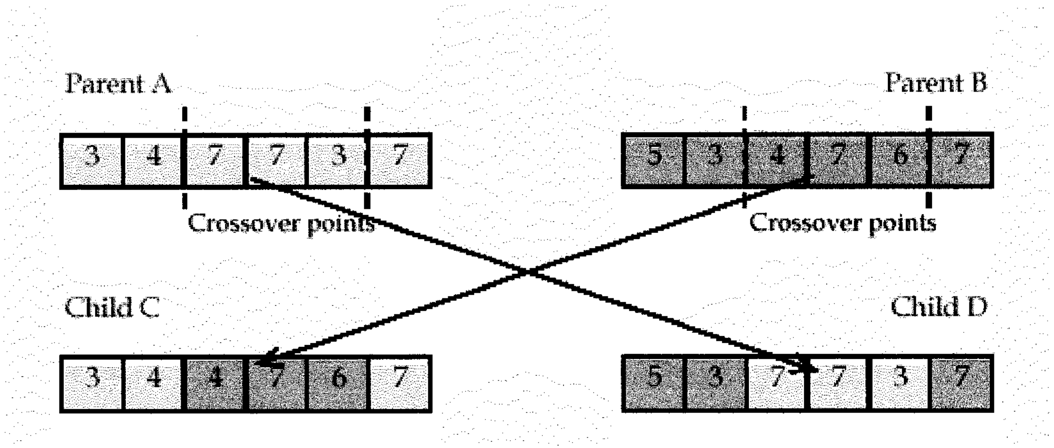
2.2.4 Crossover

After the *Selection* process, two individuals are selected randomly from the mating pool for crossover/recombination to produce two new offspring (children). The purpose of crossover is mainly to create better offspring by exchanging genetic materials between two individuals. Crossover is not usually applied to all pairs of individuals selected for mating. *Crossover Probability*, p_c, is set either experimentally or based on *schema-theorem principles* [Gol89, Sas05]. The value of p_c should be selected carefully, as it affects the performance of GA [Gaz99]. It is needed to be mentioned that, if a pattern appears in both the parents involved in a crossover, then both the newly produced offspring will also preserve that pattern [Lof06]. A random number, r, is generated between 0 to 1 and if $r \leq p_c$, the randomly selected two parents perform crossover process to produce two new offspring. Otherwise, the offspring are produced simply duplicating the parents when $r > p_c$. Typically crossover probability is considered between 0.6 to 1.0. Various types of crossover strategies have been designed for the GA, some of them are described below -

2.2.4.1 k-point Crossover : The value of *k* is usually 1 or 2, resulting in *One-point Crossover* (or *Single-point Crossover*) or *Two-point Crossover*, respectively. In one-point crossover, one crossover point is selected randomly over the parent's string length. Similarly in case of two-point crossover, two crossover points over the parent's string length are selected randomly. After the crossover points have been selected, the two parents (chromosomes) exchange their alleles (genes) over the same crossover points and produce two new offspring as shown in Fig 2.5.



(a) One-point or Single-point Crossover



(b) Two-point Crossover

Figure 2.5 : k-point Crossover

2.2.4.2 Uniform Crossover : In uniform crossover [Sys89, Spe94], the *swapping probability,* p_s, is selected before the two randomly selected parents undergo the uniform crossover. In this crossover process, every allele between the parents is exchanged according to the selected swapping probability. Fig 2.6 shows the uniform crossover between two parents A and B. Allele 2, 3 and 5 are selected from the parent, as per the swapping probability, to exchange their values, and children C and D are formed after the crossover process. The value of swapping is usually considered as 0.5 for the uniform crossover strategy.

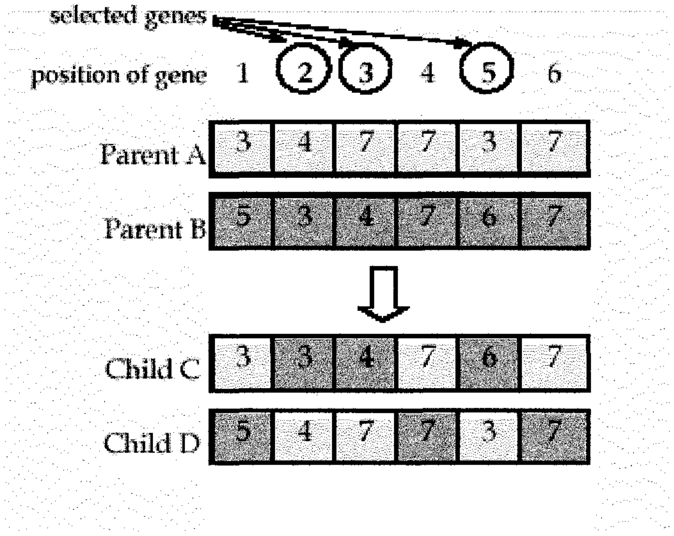


Figure 2.6 : Uniform Crossover

2.2.4.3 Mixed Crossover : In mixed crossover, the type of crossover is chosen randomly from different types of crossover methods. The types of crossover may be one-point crossover or two-point crossover or uniform crossover or any other type of crossover, but the selected crossover method remains the same for a same generation. Every type of crossover has advantages and disadvantages. The mixed crossover does not care about any special structure of the individuals nor does it care about the problem itself. At the beginning, random crossover assigns a unique constant to each type of crossover. Every time a crossover is executed, a number is randomly picked among the assigned constant and the corresponding crossover is then executed.

2.2.5 Mutation

Mutation is applied to each child individually after crossover, to achieve a better fitness value. The mutation process also restores lost genetic values when the population *converges* too fast [Gol89]. If two parents, having same allele, are selected for mating, then the crossover will not change the structure of the new child. Mutation is designed to overcome such problems. It changes each allele value according to the *Mutation Probability* p_m. *Bitwise Mutation* is a common type of mutation, where each bit in a binary string is altered with a probability of p_m. Fig 2.7 illustrates a bitwise mutation where randomly selected *mutation point* is 6 and the value of allele (gene) is 0. After the mutation the value of same position for the child becomes 1.

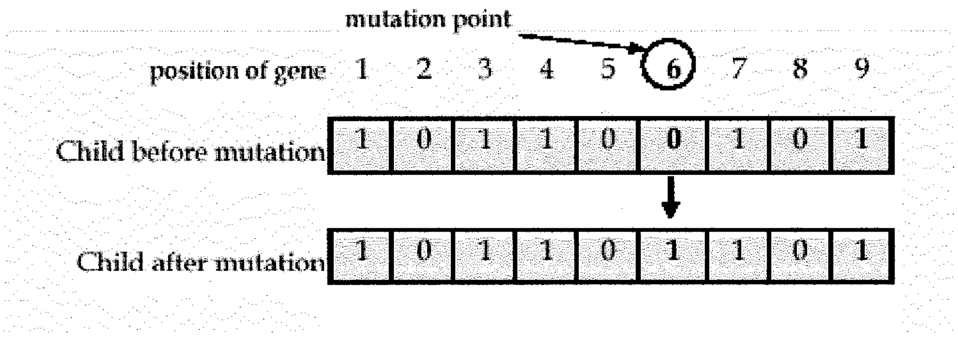


Figure 2.7 : Bitwise Mutation

Similar to crossover probability, mutation probability also influences the GA performance. If the value of the mutation probability is set too low, the GA will converge early, before reaching an acceptable solution. On the other hand, it may cause a stability problem in population if the value of p_m is set too high [Gaz99].

2.2.6 Replacement

After selection, crossover, and mutation, new offspring are produced, and the population for the next generation must be prepared. A certain percentage, or all of the offspring produced in the previous generation, can be selected for the next generation. Different types of replacement strategies have been proposed in the GA literature –

2.2.6.1 Delete-all : In this strategy, all parents are deleted and all newly produced offspring form the next generation. To implement, we can consider an empty population *P* and then every time new offspring are created they are put into *P*. At the end, we just delete all the population from the previous generation and *P* becomes the new generation.

2.2.6.2 Elitism : Cossover and mutation cannot guarantee creating of a more fit child than the parents in the previous generation. This is the disadvantage of the *Delete-all* replacement strategy. To overcome this problem, some of the elite parents from the previous generation are included into the next generation by replacing the same number of worse fit offspring. This is the most common replacement and known as Elitism strategy. Elitism ensures that the best fitness chromosomes always survive to the next generation, while the worst are not carried to the succeeding generation. To implement this strategy, we need to fix the percentage of most elite individual that will be inserted into the next generation and probability of elitism in each generation.

2.2.7 Stopping Criterion

The stopping criterion is used to stop the execution of the GA. The GA does not guarantee to find the global optimal solution of a given complex problem, but an acceptable solution can be obtained within a reasonable time. A satisfactory stopping criterion can be obtained –

1. When an acceptable solution is found that satisfies the minimum criteria,
2. When the number of generations excees the value of Gen_max, as defined by the user,
3. When the population is *Converged*, i.e. the average fitness of the population is at least 95% of the best fitness, or
4. Any combination of the above rules

If the criterion 2 for convergence is applied, the probability of mutation should be increased to introduce more diversity into the population [Liu02]. In order to achieve a better solution with less execution time, the stopping criterion should be carefully selected.

CHAPTER 3

Genetic Algorithm Based Routing

In this chapter, we will describe our tow-tiered network model and outline our GA based algorithm for routing in the upper tier relay node network. The objective of the proposed GA is to find an optimal schedule for data gathering in a two-tiered WSN such that the lifetime of the network is maximized.

3.1 Network Model

We consider a two-tiered wireless sensor network with n relay nodes, labelled as node numbers 1, 2, 3, ..., n and one base station, labelled as node number n+1. Each sensor node belongs to only one cluster and each cluster has one relay node that acts as a cluster head of that cluster. In other words, let D be the set of all sensor nodes which form total m clusters, and D_i, $1 \leq i \leq m$, be the set of sensor nodes belongs to the i^{th} cluster. Then, $D = D_1 \cup D_2 \cup \ldots \cup D_m$ and $D_i \cap D_j = \emptyset$, for $i \neq j$.

The sensor nodes transmit their data directly to their respective relay node belongs to that cluster, and relay nodes of each cluster forwards the gathered data toward the base station. Each relay node i transmits data to one other node j (either another relay node or the base station), such that node j is within the transmission range of node i and is closer to the base station than node i. The base station only receives data and there is no transmission from base station to any relay node. The initial energy provisioned in each relay node is assumed to be equal. The amount of bits received by each relay node in a round is fixed, but may vary from one node to another.

In this paper, we adopt the "non-flow-splitting" model, for the upper tier relay node network. In this model, the outgoing data from each relay node flows to only one destination node (either another relay node or base station). It cannot be split into a number of different flows to be sent different destination nodes simultaneously. The *non-flow-splitting* routing model has been investigated in the papers [Bari06a, Bari06b, Hou04].

The routing schedule is computed here by some centralized entity (e.g. the base station), which is not power constrained. It is assumed that all nodes are stationary after deployment, and that relay nodes are either placed at specified locations (determined by a suitable placement strategy such as in [Bari06b]), or can report their current location for initial configuration. We assume that the average data generated by each sensor node and the distribution of the sensor nodes into their clusters is known. It is assumed that the initial energy provisioned in each relay node is equal. As already mentioned, the routing schedule is calculated centrally, based on the positions and expected data rates at each relay node. The computed routing schedule may either be pre-loaded on to the nodes prior to the deployment of the network, or may be broadcast to all nodes during the setup phase. We note that, in the second scenario, the amount of data exchanged for setting up the initial configuration is very small and its effect on the total lifetime of the network is negligible. Each period of data gathering is referred to as a *round* [Bari06a, Bari06b, Kal02], and the lifetime is measured by the number of rounds until the first relay node runs out of power. The relay nodes are placed uniformly over the sensing area, using the placement strategy outlined in [Bari06b].

In the following sections, we will outline the main steps needed to implement our genetic algorithm for energy efficient routing.

3.2 Chromosome Representation

The chromosome corresponding to a specific routing scheme is represented here as a string of node-numbers. The length of each chromosome is always equal to the number of relay nodes in the network. The *position* of each cell (gene) represents a source relay node number – which is transmitting data. The cell *value* represents the destination relay node number – i.e. the node which is directly receiving the data from the corresponding source node. The base station only receives data from relay nodes and does not transmit data. Therefore, it is not necessary to represent it explicitly in the chromosome as a *source* node. The relationship between a specific routing scheme and the corresponding chromosome is illustrated in Figure 3.1 for a network of size 7, with 6 relay nodes and 1 base station. The routing scheme can be represented by the chromosome of length 6 (corresponding to the 6 relay nodes). Fig. 3.1(a) shows the logical routing path for such network and the corresponding chromosome is shown in Fig. 3.1(b). In Fig. 3.1(b), the value of the gene in position 1 is 3, indicating that node 1 transmits to node 3. Similarly, the value in position 3 is 7, indicating that node 3 transmits to node 7 (base station), and so on.

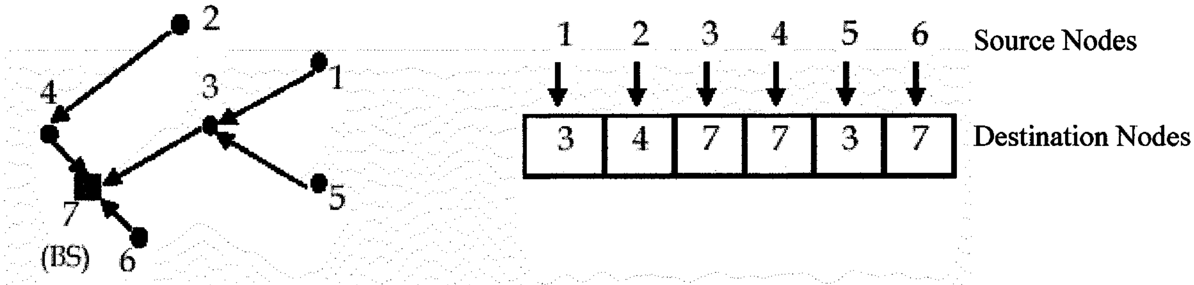


Fig. 3.1 : (a) A routing scheme and (b) corresponding chromosome representation for a 7-node network

3.3 Generating Random Graphs

To generate random graphs for an initial population, the following network flow constraints should be maintained :

1. Each relay node receives data from the sensor nodes belonging to its own cluster, and can also receive data from any number of other relay nodes.
2. Each relay node *i* transmits data to one other node *j* (either another relay node or the base station), such that node *j* is within the transmission range of node *i* and is closer to the base station than node *i*.
3. The base station only receives data, and there is no transmission from base station to any relay node.

For the network of size n+1, an integer array of size n is first initialized by setting all values to 0, to indicate no links currently exist in the network. First select node no. 1 as the source node and generate a random number j between 1 to n+1 for the destination node that satisfies the above flow constraint no. 2. Replace the value of cell no.1 from 0 to j – which indicates j is the destination node for node no. 1. Then, consider j as the source node (when j is not base station) and generate another random number k (destination node for j) between 1 to n+1 satisfying the mentioned flow constraint. If k is not the base station, then repeat the process until the base station is found as the destination node. Special care should be taken selecting destination nodes so that the flow does not create any disconnected graph. The node j can be selected as the destination node for the source node i where base station is n+1 such that :

i) $i \neq n+1$

ii) $j \in \{1 \ldots n+1\}$ and $i \neq j$

iii) $d_{j,\, n+1} < d_{i,\, n+1}$

iv) $d_{i,\, j} \leq$ data transmission range , where $d_{i,\, j}$ is the Euclidean distance from node i to node j.

After receiving the base station as the destination node, select the next available source node starting from the left side of the array whose destination node is still labeled as 0 (i.e. no link yet established from that node) and follow the above procedure considering the node as the source node. If the generated destination node is found as any source node selected in earlier process, replace 0 by the new destination node and find the next available source node to determine the flow from the node. This cycle will be continued until all the source nodes find their respective destination nodes. To generate a random graph of size 7 as shown in Fig 3.1, the following steps were made :

1) 1 → 3, 3 → 7 (found base station, select node 2 as source node for the next flow)

2) 2 → 4, 4 → 7 (found base station, select node 5 for next flow as 3 and 4 already selected for source nodes)
3) 5 → 3 (3 is selected as source node for the flow from node 1)
4) 6 → 7 (found base station, and since destination nodes for all source nodes are found, generating a random graph is completed)

For a population of size *P*, we need to generate such *P* numbers of randomly generated graphs following the above procedure and calculate the fitness value of each individual before perform the initial selection.

3.4 Fitness Function

Each new individual needs to be evaluated to determine its fitness value. The fitness value is nothing but the lifetime of the network, represented by the total number of rounds until the first relay node runs out of battery power. The value can be computed for an individual through the use of the following equation [Bari06a],

$$L_{net} = \frac{E_{initial}}{E_{max}} \qquad (1)$$

where, L_{net} is the network lifetime in terms of rounds, and $E_{initial}$ is the initial energy of a relay node, which is known beforehand. E_{max} is the maximum energy per round dissipated by any relay node in the individual, for the routing scheme defined by the chromosome. It is defined as,

$$\mathrm{E}_{\max} = \max(E_x, 1 \le x \le n) \qquad (2)$$

where, E_x is the total energy dissipated by relay node x for one round of data gathering. E_x is obtained by summing the total energy dissipation for transmitting data (E_{Tx}), and the total energy dissipation for receiving data (E_{Rx}) for the relay node x, i.e.

$$E_x = E_{Tx} + E_{Rx} \tag{3}$$

The total transmitting energy and receiving energy can be computed based on the first order radio model as mentioned in [Hei00]. In this paper, the authors assumed a simple model radio (Fig. 3.2) that dissipates 50 nJ/bit to run the transmitter or receiver and 100 pJ/bit/m^2 (for the *path loss exponent*, $m = 2$) to run the transmit amplifier.

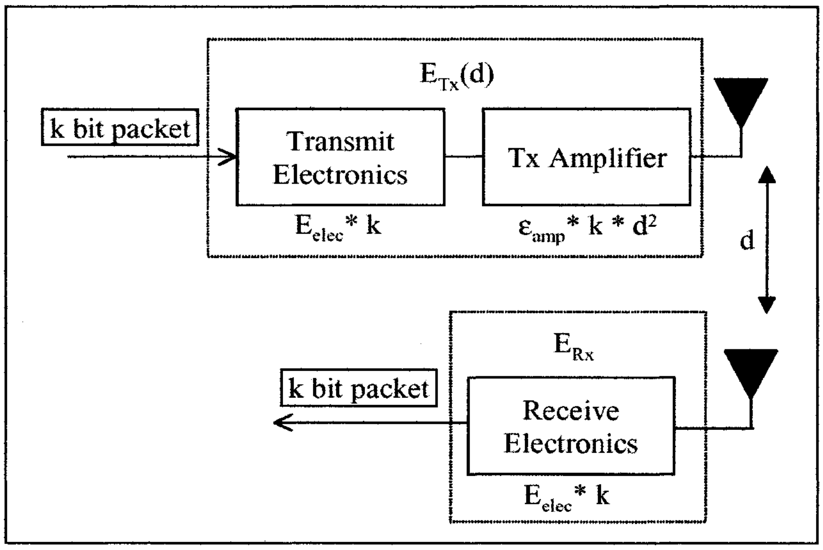


Figure 3.2 : First Order Radio Model (where, m=2)

The transmitting energy, E_{Tx}, is dissipated in a round by each relay node x, $1 \le x \le n$, to transmit k amount of data to another node y, $1 \le y \le n+1$, can be defined as,

$$E_{Tx} = E_{elec} * k + \varepsilon_{amp} * k * d^m_{x,y} \tag{4}$$

where, $d_{x,y}$ is the Euclidian distance between node x and y, E_{elec} is the transmit energy coefficient, ε_{amp} is the amplifier coefficient and m is the path loss exponent. The value of m is considered as 2 for the free space, when using short to medium–range radio communication [Pan03, Bari06a].

Similarly, the received energy (E_{Rx}) dissipated in a round by each relay node x, $1 \leq x \leq n$, can be defined as,

$$E_{Rx} = E_{elec} * k \qquad (5)$$

where, k is the number of bits received by a relay node in a round, and E_{elec} is the receive energy coefficient.

3.5 Selection of Individuals

Selection of individuals is carried out by using Roulette-Wheel selection method [Hol75, Gol89, Sas05], where the probability of being selected increases with the fitness value of the individual chromosome. In Roulette-Wheel selection, each individual in the population is assigned a roulette wheel slot sized in proportion to its fitness. That is, in the biased roulette wheel, good solutions have a larger slot size than the less fit solutions.

For n individuals ($x_1 \ldots x_n$), the Roulette-Wheel selection scheme can be implemented as follows,

1. Evaluate the fitness value, f_i, for each individual x_i, where $1 \leq i \leq n$ and n is the total number of relay nodes. This is the L_{net} value as discussed in section 3.4
2. Calculate population total fitness value, $f_T = \sum_{i=1}^{n} f_i$
3. Find the individual probability (i.e. slot size), $p_i = \frac{f_i}{f_T}$
4. Calculate the cumulative probability, q_i, for each individual as $q_i = \sum_{j=1}^{i} p_j$
5. Generate a uniform random number, $r \in (0, 1]$
6. If $r < q_1$ then select the first individual, x_1, else select the individual x_i such that $q_{i-1} < r \leq q_i$
7. Repeat steps 5–6 n times to select n candidates in the mating pool.

To illustrate the above procedure let us consider a population with five individuals ($n = 5$), with the fitness values as shown in the table below. The total fitness, $f_T = \sum_{i=1}^{n} f_i =$ 4+2+7+4+3 = 20. The probability of selecting an individual and the corresponding cumulative probabilities are shown in Table 3.1 below,

Individual	**A**	**B**	**C**	**D**	**E**
Fitness, f_i	4	2	7	4	3
Probability, p_i	4/20 = 0.20	0.10	0.35	0.20	0.15
Cumulative probability, q_i	0.20	0.30	0.65	0.85	1.00

Table 3.1 : Calculation of cumulative probability for Roulette-Wheel Selection

Now generate a random number r such that $r \in (0, 1]$. Let the value of r be 0.58, then the third individual, **C**, will be selected since $q_2 = 0.30 < 0.58 \leq q_3 = 0.65$. If value of r is 0.14, then first individual **A** will be selected since $0.14 < q_1=0.20$.

3.6 Mixed Crossover

The *crossover* operation is performed in order to produce new offspring from the selected parents. The type of crossover is selected randomly from the following:

i) uniform crossover (with swapping probability 0.5) and

ii) k-point crossover (k = 1, 2, or 3, selected randomly) for each generation.

The different types of crossover schemes are discussed in detail in Section 2.2.4. Fig. 3.4 shows an example of 2-point crossover with two parent chromosomes, Parent A & Parent B, in a network with 6 relay nodes (n=6) and one base station. Two random numbers r_1 and r_2 ($r_1 \neq r_2$) are selected such that $r_1, r_2 \in \{1 \ldots n\text{-}1\}$ to determine the crossover points. Let, r_1=2 and r_2 =5 represent the crossover points as indicated by the dotted lines in Fig 3.4.

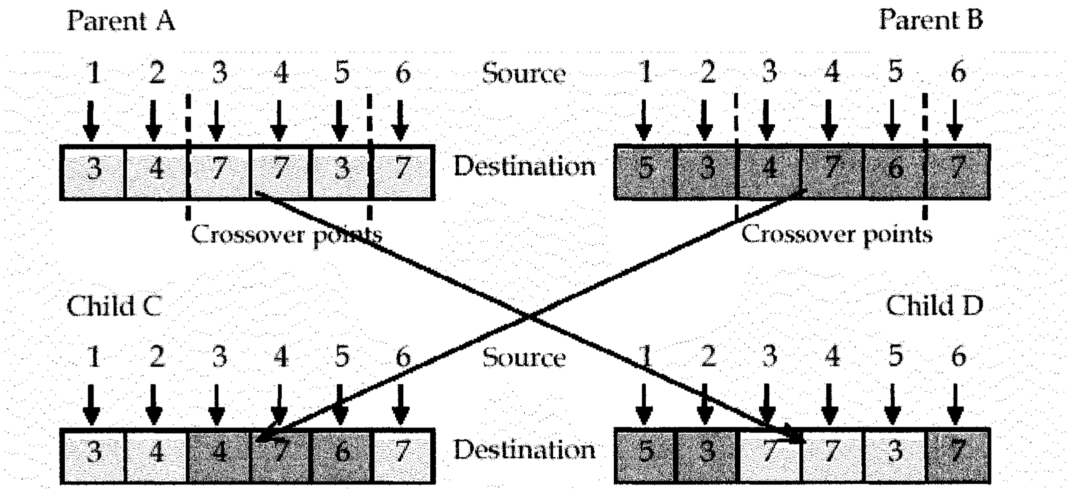


Figure 3.4 : 2-points crossover in a network of size 7

After the crossover, two new child chromosomes, Child C & Child D are generated by interchanging their gene values of the parents at the crossover point r_1 and r_2. Our crossover scheme guarantees that the two offspring resulting from the crossover will always be valid solutions, corresponding to a specific non-flow-splitting routing scheme. The rate of crossover should be carefully designed, as it affects the GA performance greatly.

3.7 Network Mutation

In GA, *mutation* is usually applied, on a randomly selected gene, after the process of crossover, to improve the fitness value of individual. In the proposed GA, instead of randomly selecting a gene for mutation, we identify the *critical node*, which becomes the selected candidate for the mutation operation. The critical node is the node which dissipates maximum energy (due to the receiving and/or transmitting of its data), and hence determines the lifetime of the network. The purpose of selecting the critical node for mutation is to reduce the total energy dissipation by the critical node, with the expectation that this is likely to increase the network lifetime. Energy dissipation of the critical node can be reduced in one of two ways :

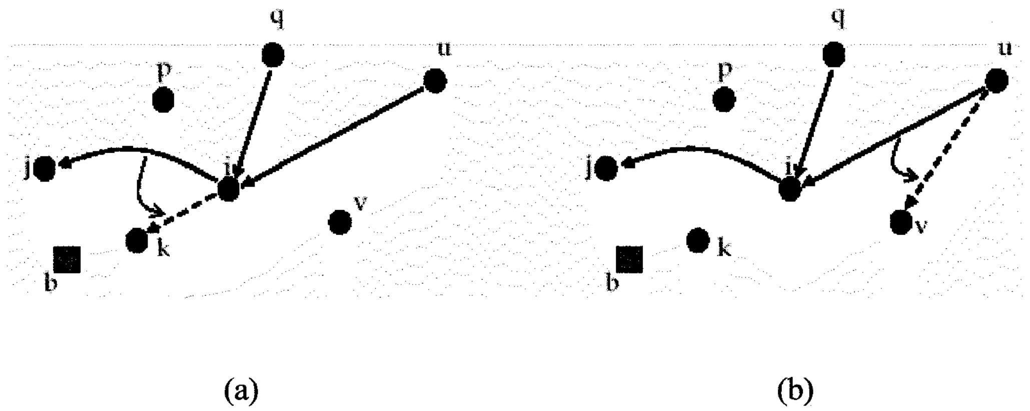


Figure 3.5 : Redistribution of load on critical node in mutation process (only the affected flows are shown here)

i) by changing the destination of the critical node to a randomly selected suitable node which is closer to the critical node (Fig. 3.5(a)), or

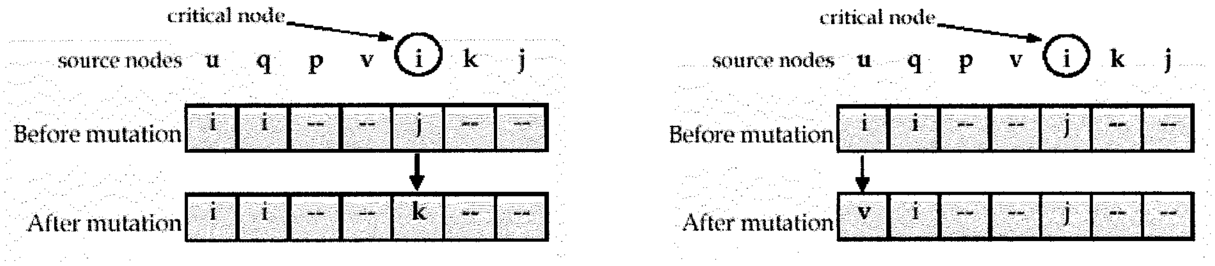
ii) by diverting some incoming flow away from the critical node (Fig. 3.5(b)).

The first option reduces the *distance* that the critical node has to transmit and the second option reduces the *amount of data* that the critical node has to transmit.

Fig. 3.5 shows a portion of a network where i is the critical node and b is the base station. As shown in the Fig. 3.5(a), changing the destination node for the critical node i, from j to k should be done in a way such that $d_{i,k} < d_{i,j}$ and $d_{k,b} < d_{i,b}$, where $d_{x,y}$ is the

Euclidean distance from node x to node y. Similarly, in Fig. 3.5(b), we reduce the load on the critical node i, by diverting the traffic from node u, so that it is sent to node v, instead of node i.

The alternate destination node v should be selected such that $d_{v,b} < d_{u,b}$ and v lies within the transmission range of u. The changes in the corresponding chromosome representations are depicted in Figures 3.6(a) and 3.6(b), respectively.



(a) (b)

Figure 3.6 : Chromosome representation of networks after mutation performed

The rate of mutation is also designed carefully, and it is closely related to the rate of crossover [Gaz99]. If the mutation rate is set too low, the GA will converge too early before the optimal solution is found, and if it is set too high, it may cause stability problem in the population.

3.8 Elitism and Stopping Condition

We have used *elitism* in selecting the individuals the next generation. 10% of the most elite (highest fitness value) individuals from the previous generation are inserted into the new generation, replacing the lowest 10% of individuals in the current generation. This process is implemented with a probability of 0.80 for each generation. This means that there is 80% chance that elitism will be used to select individuals for the next generation. A pre-specified maximum number of generations is set in the proposed GA, depending on the network size, to stop the program when the population does not converge.

CHAPTER 4

Experimental Results and Performance Analysis

In this chapter we will present and analyze our experimental results. We have run a number of experiments with different networks, ranging in size from 11 nodes to over 300 nodes. We have also experimented with different positions of the base station, as well as uniform and non-uniform data rates for the relay nodes. We will first describe the network parameters and GA parameters used in our experiments and discuss the results from each set of experiments.

The numbers of sensor nodes for each cluster are not assumed any here, but a fixed receiving data rate (1000 *bits/round*) for each relay node has been assumed as mentioned in [Bari60a]. A random data receiving rate then again considered, varying from 500 to 2000 *bits/round*, to find out the 95% confidence level for each network for each placement of Base Station. The communication energy dissipation is based on the first order radio model as mentioned in [Hei00].

4.1 Experimental Setup

4.1.1 Network Parameters

We consider a two-tier sensor network architecture, with the relay nodes acting as cluster heads and the sensor nodes transmitting their data directly to their respective cluster heads. The relay nodes use multi-hop routing to transmit their data to the base station. The positions of the relay nodes are determined using the strategy given in [Bari06b], by covering the sensing area with an imaginary grid and placing relay nodes at predetermined locations. The number of required relay nodes depends on the sensing area. In general a larger number of relay nodes are used, as the sensing area is increased. The network parameters, including the sensing area and the grid size, for each of our networks, are shown below in Table 4.1.

Network Size	Nos. of Relay Nodes (Nos. of Clusters)	Grid Dimension (m x m)	Network Area (m x m)
11	10	40 x 40	160 x 120
13	12	40 x 40	160 x 160
16	15	40 x 40	200 x 160
18	17	40 x 40	240 x 160
20	19	40 x 40	240 x 200
22	21	40 x 40	240 x 200
25	24	40 x 40	240 x 240
29	28	40 x 40	280 x 240
32	31	40 x 40	320 x 240
36	35	40 x 40	360 x 240
41	41	40 x 40	360 x 280
45	44	40 x 40	400 x 280
50	49	40 x 40	400 x 320
55	54	40 x 40	440 x 320
61	60	40 x 40	440 x 360
85	84	40 x 40	480 x 480

Table 4.1 : Sensor networks and their sizes

For each relay node, we use the following values to characterize the energy dissipation per bit [Hei02], as well as the transmission range and the initial energy [Tan06].

a) Receiver consumption rate, $E_{elec} = 50$ *nJ/bit*

b) Transmitter consumption rate, $E_{elec} = 50$ *nJ/bit*

c) Amplifier (to transmit data) consumption rate, $\varepsilon_{amp} = 100$ $pJ/bit/m^2$

d) The data transmission range of each relay node = 200 *m* [Tan06]

e) Initial energy of each relay node = 5 *Jouls* [Tan06]

We also experimented with different values of the path loss exponent *m*, uniform and non-uniform data rates for the relay nodes, as well as two different positions of the base station – i) base station placed at one corner of the sensing area and ii) base station placed at the middle of one of the boundary edges of the sensing area

4.1.2 GA Parameters

The scalability and performance of the genetic algorithm is greatly affected by a number of factors, such as the population size, method and the rate of crossover and mutation, and the method of replacement. A small population size may lead to premature convergence before reaching an acceptable solution. On the other hand, if the population size is too large, it leads to unnecessary computations [Sas05].

A number of experiments have been run with different values of these parameters to determine the optimal set for each of the mentioned 16 sizes of networks. Increasing the rate of crossover (and mutation) allowed the GA to reach the optimal solution faster. Initial selection was performed by only selecting individuals having a fitness value that is above a predetermined threshold. The probability of elitism in each generation was set to 1 for small networks and to 0.80 for larger networks. Although the GA was allowed to run for a maximum of 100 generations, the best solution was typically found within 20 to 30 generations. Table 4.2 below shows the values of the different GA parameters corresponding to each of the networks considered in our experiments.

Network Size	Population Size	Max Generation	CrossOver Type	Rate of CrossOver	Rate of Mutation	Elitism Probability
11	50	10	Mixed	1	0.6	1
13	100	30	Mixed	1	0.6	1
16	800	100	Mixed	1	0.6	1
18	5,000	100	Mixed	1	0.8	0.8
20	5,000	100	Mixed	1	0.8	0.8
22	10,000	100	Mixed	1	1	0.8
25	10,000	100	Mixed	1	1	0.8
29	10,000	100	Mixed	1	1	0.8
32	10,000	100	Mixed	1	1	0.8
36	10,000	100	Mixed	1	1	0.8
41	10,000	100	Mixed	1	1	0.8
45	10,000	100	Mixed	1	1	0.8
50	10,000	100	Mixed	1	1	0.8
55	10,000	100	Mixed	1	1	0.8
61	10,000	100	Mixed	1	1	0.8
85	10,000	100	Mixed	1	1	0.8

Table 4.2 : GA parameters set for different sizes of networks

4.2 Results with Base Station at a Corner

Our first set of experiments was carried out with the base station positioned at the lower left corner of the sensing area. This sub-section describes the results corresponding to this placement (shown in Fig. 4.1).

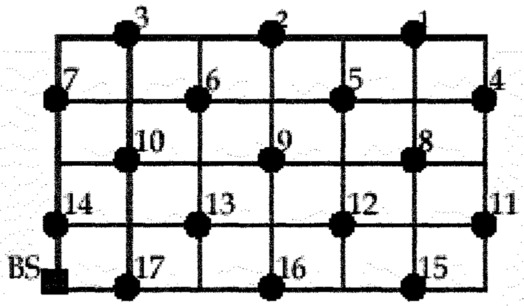


Figure 4.1 : 18 Node Model Sensor Network
(17 relay nodes and 1 Base Station)

The data received by the relay nodes were generated in one of two ways:

i) Uniform data receiving rate (UDRR): Each relay node receives the same amount of data (1,000 bits/round) from the sensor nodes in its cluster, and

ii) Non-uniform data receiving rate (NDRR): The amount of data generated by the sensor nodes varies from cluster to cluster. The data rate for each cluster is selected randomly from 500 bits/round – 2,000 bits/round.

4.2.1 Uniform Data Receiving Rate (UDRR)

The optimal lifetimes found for the networks by our proposed GA based approach are compared with those obtained through traditional multi-hop routing schemes. These include MTEM (Minimum Transmission Energy Model) [Gup03], where each relay node transmits data to its nearest neighbor node, such that the destination relay node is closer to the base station than the source relay node, and MHRM (Minimum Hop Routing Model) [Hei00] where each relay node finds a path to the base station that minimizes the number of hops. The results, shown in Fig 4.2, indicate that the GA is able to improve network lifetime by almost 200% compared to the traditional routing schemes.

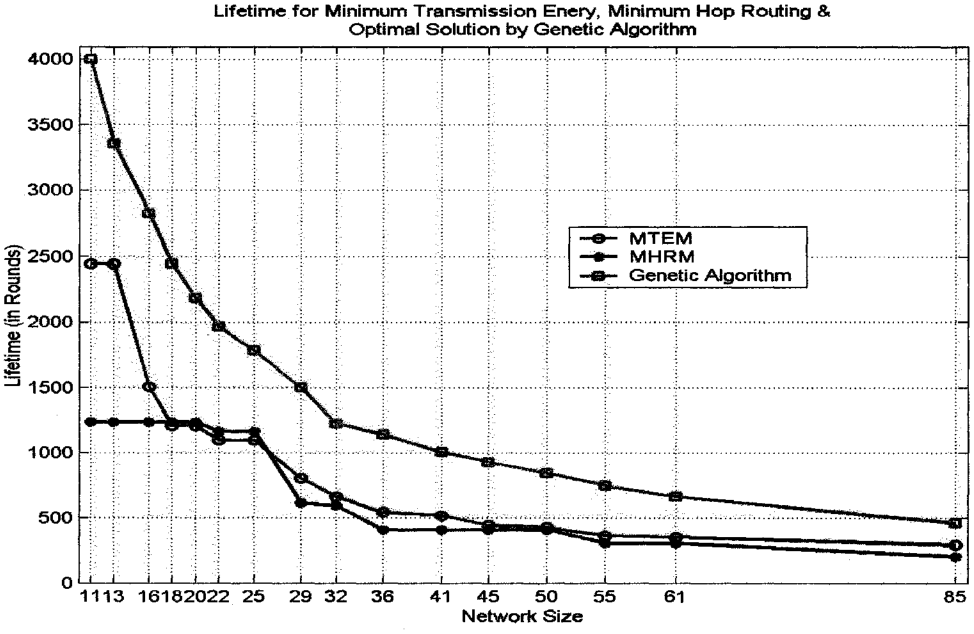


Figure 4.2 : Comparison of the lifetimes achieved by GA

to that of using conventional methods

For smaller networks (up to 18), the ILP formulation proposed in [Bari06a] is able to determine the optimal solution of the relay node networks. The same optimal lifetimes can also be obtained through the use of the GA with much less time. For example, in the case of a 18-node network, the proposed GA found the optimal solution in less than 25 sec, whereas the said ILP required several hours, using the solver ilog CPLEX [Cplex91].

Fig 4.3 shows a comparison of the solution times required using GA and CPLEX. One important factor, affecting the execution time, is the number of nodes in the input graph. As the size of networks increase, the time required to prepare a complete generation takes more time. In the basic genetic algorithm, the crossover rates and the mutation rates have negligible effect on the running time [Gaz99]. Fig 4.4 shows each generation times (in *msec*) for different sizes of the network.

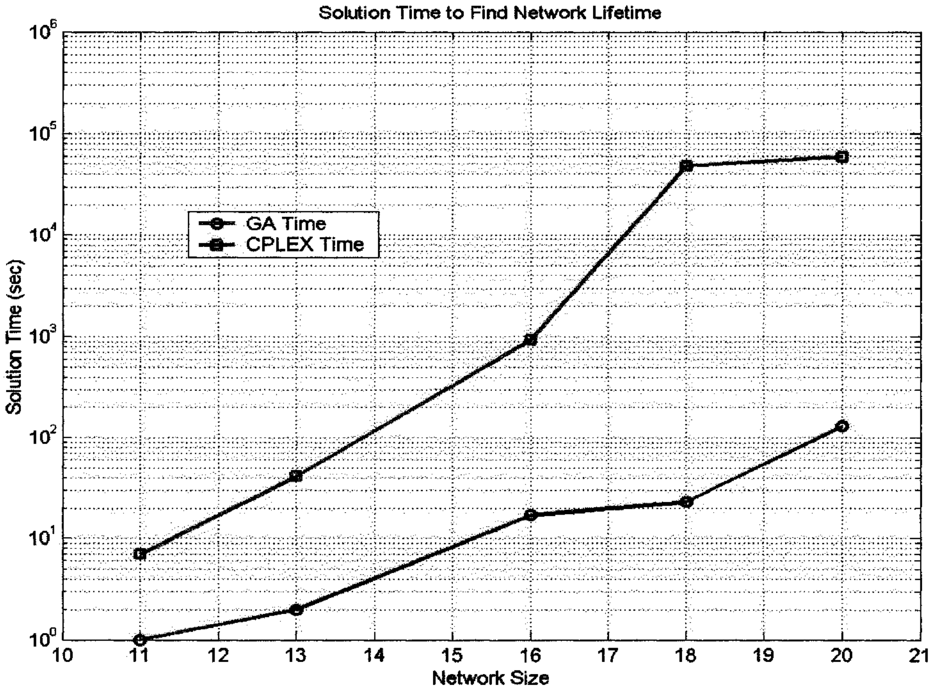


Figure 4.3 : Comparison of solution times for GA and CPLEX

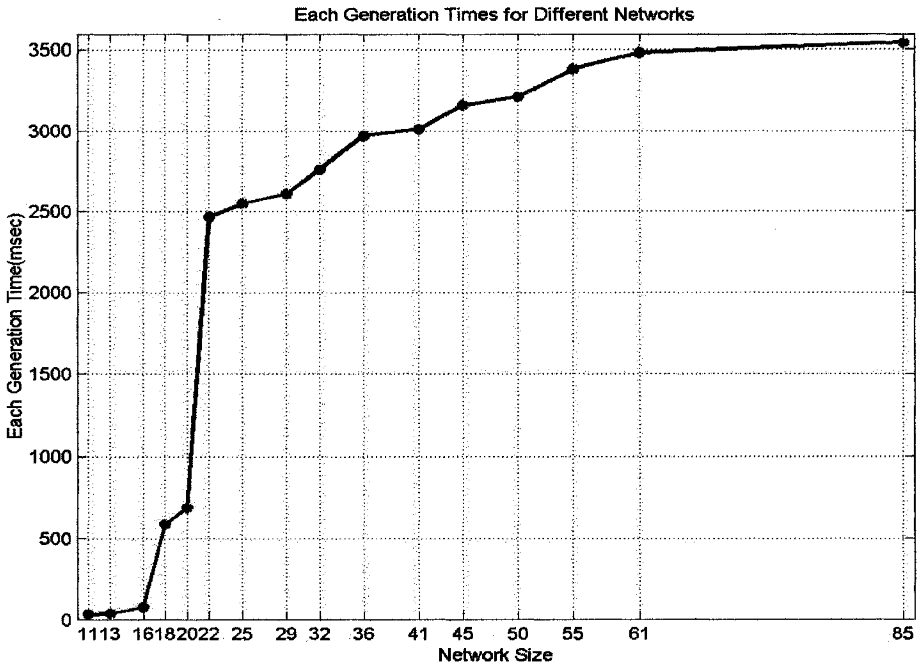


Figure 4.4 : Generation time for different networks

4.2.2 Non-uniform Data Receiving Rate (NDRR)

In the previous section, data receiving rates for all relay nodes were equal (1,000 *bits/round*). In this section, we randomly assigned different data rates to the clusters (ranging from 500 to 2,000 *bits/round*). For each network, the GA is run 5 times (with different values of data receiving rates), and the average lifetimes are plotted in Fig. 4.5 below. Overall, the GA improves the network lifetimes more than 250% over MHRM and around 200% over MTEM routing schemes.

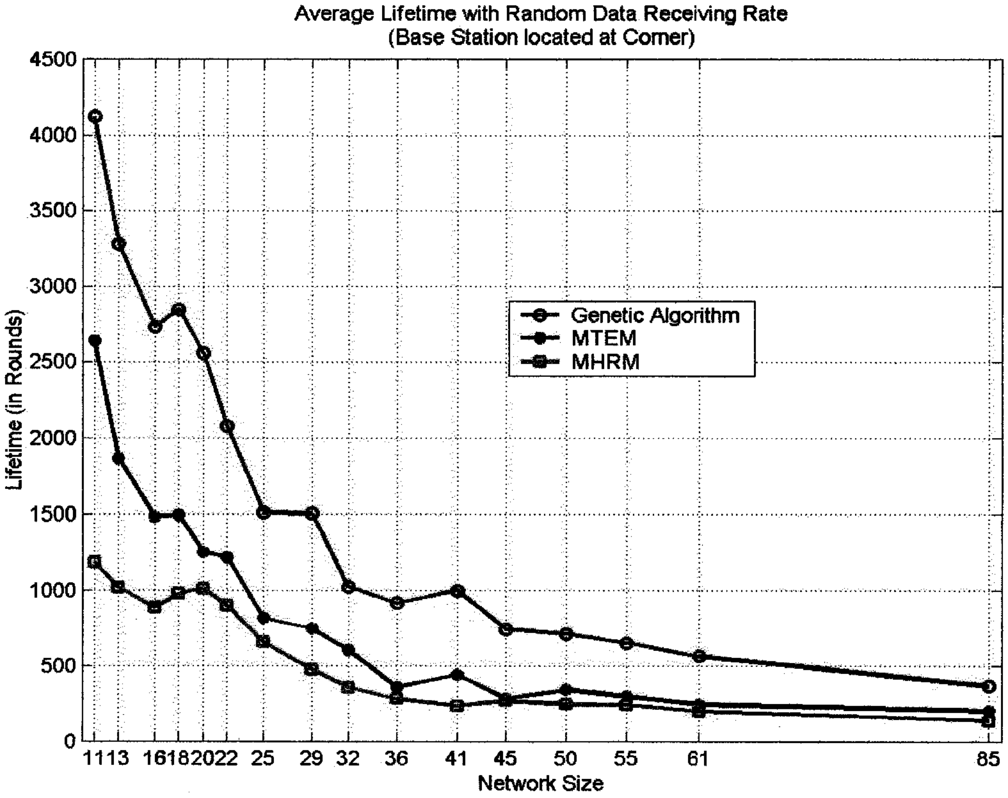


Figure 4.5 : Comparison of the lifetimes considering NDRR

The network lifetimes obtained using uniform and non-uniform data receiving rates are almost same, as indicated below in Fig. 4.6.

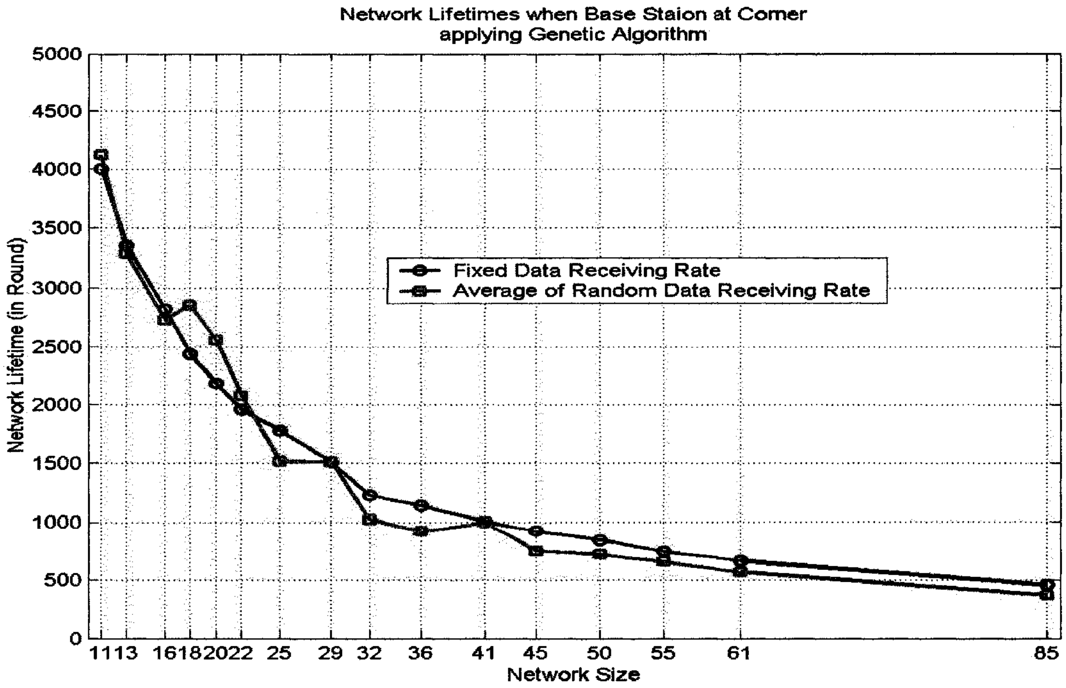


Figure 4.6 : Comparison of the lifetimes considering UDRR and NDRR

Table 4.3 on the next page shows the 95% Confidence Interval (C.I.) chart with upper bound and lower bound of network lifetimes (in *number of rounds*) found using GA, and the corresponding improvements compared to MHRM and MTEM routing respectively. For example, for a 11-node network, the 95% C.I. is between 273 and 463 for MHRM network. This means, we can state with 95% confidence level that the mean of percentage improvement of the GA compared to MHRM will not be less than 273 and will not be more than 463.

Network Size	Network Lifetime with 95% Confidence Interval					
	Lifetime (in *Rounds*) obtained in GA		Percentage Improvement over Minimum Hop Routing Transmission (MHRM)		Percentage Improvement over Minimum Transmission Energy Model (MTEM)	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
11	3,622	4,618	273	463	140	181
13	2,952	3,615	278	387	166	187
16	2,232	3,224	242	380	140	256
18	2,319	2,855	229	318	181	314
20	2,360	2,754	223	291	177	247
22	1,983	2,172	220	244	151	198
25	1,393	1,628	202	261	151	241
29	1,284	1,726	264	387	159	272
32	877	1,165	218	373	139	203
36	860	970	303	342	225	289
41	902	1,094	368	476	181	302
45	605	892	187	429	210	327
50	586	850	252	335	191	225
55	575	737	215	336	196	248
61	533	602	238	339	196	276
85	336	402	200	368	158	223

Table 4.3 : Statistical Analysis of Improvement of Network Lifetime using GA
(when Base Station at Corner of Network Area)

4.3 Results with Base Station at the Middle of Left Boundary

The previous set of experiments was carried out placing the base station at the corner of the imaginary network grid. In this section, we change the position of the base station, placing it at the middle of left edge of the sensing area (as shown in Fig. 4.7). As before, we consider both uniform and non-uniform data rates for the cluster heads.

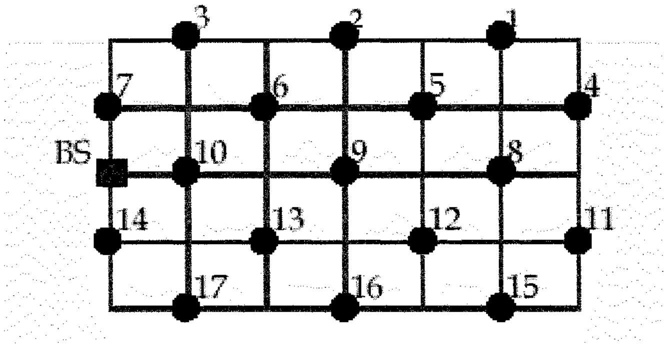


Figure 4.7 : 18 Node Model Sensor Network with BS at the Middle-Left Boundary

4.3.1 Uniform Data Receiving Rate (UDRR)

In this case, each relay node receives data at a rate of 1,000 *bits/round.* The results for our GA and for traditional routing schemes are shown in Fig 18. We note that the GA is able to improve network lifetime an average of 255%, compared to both METM and MHRM. We also observe that this placement of the base station improves the network lifetime compared to that of placing the base station at corner of the network, i.e. at the coordinate (0,0). For example, for a 18 node network, the network lifetime was found 2,440 when the base station was located at corner (Fig. 4.1), and for the same network the lifetime becomes 3,030 due to placing the base station at mid-left edge of the network area.

Figure 4.9 shows a comparison of the network lifetimes achieved using our GA, for each of the base station placement options.

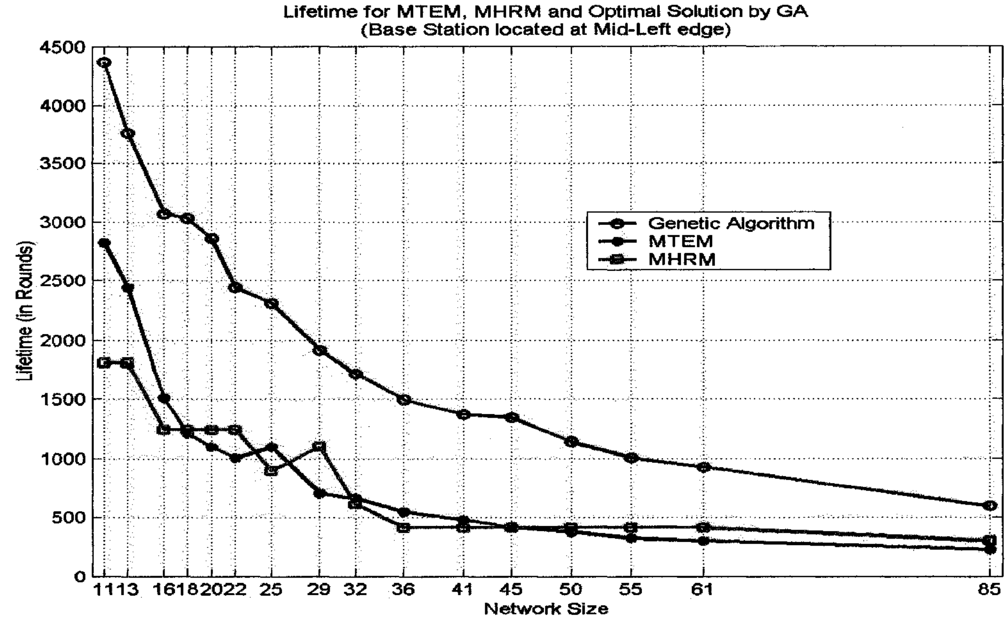


Figure 4.8 : Comparison of network lifetimes – Uniform data receiving rate (Base Station is located at mid-left edge)

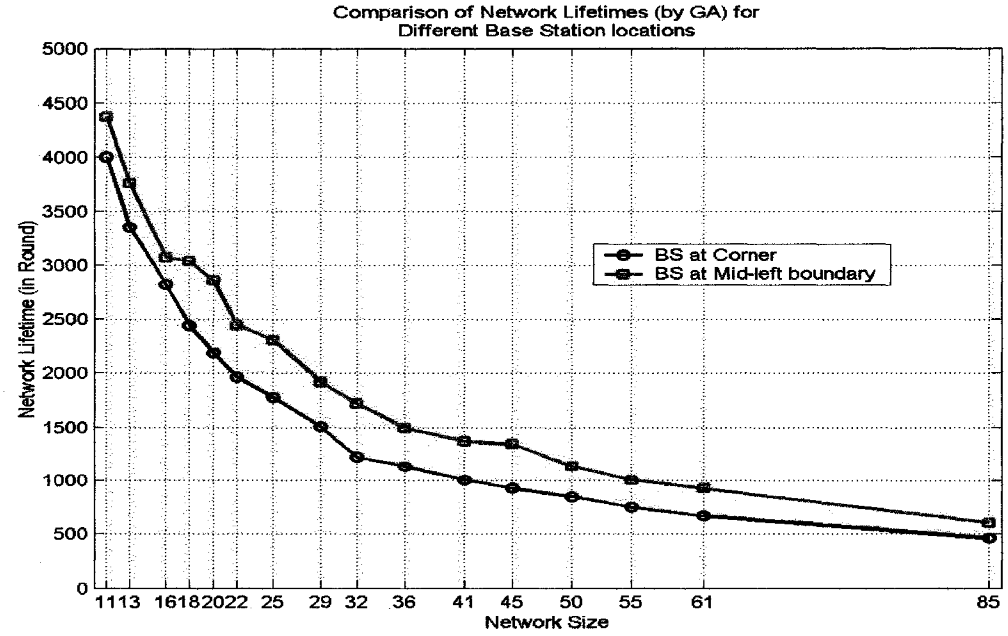


Figure 4.9 : Comparison of network lifetimes obtained by GA at UDRR for different locations of base station

4.3.2 Non-uniform Data Receiving Rate (NDRR)

The data rate for each cluster is selected randomly between 500 bits/round ~ 2,000 bits/round. The average results found from 5 runs of this experiment are shown in Fig. 4.10 below. The GA clearly outperforms the traditional routing schemes, improving the network lifetimes by almost 300% over MHRM as well as over MTEM routing.

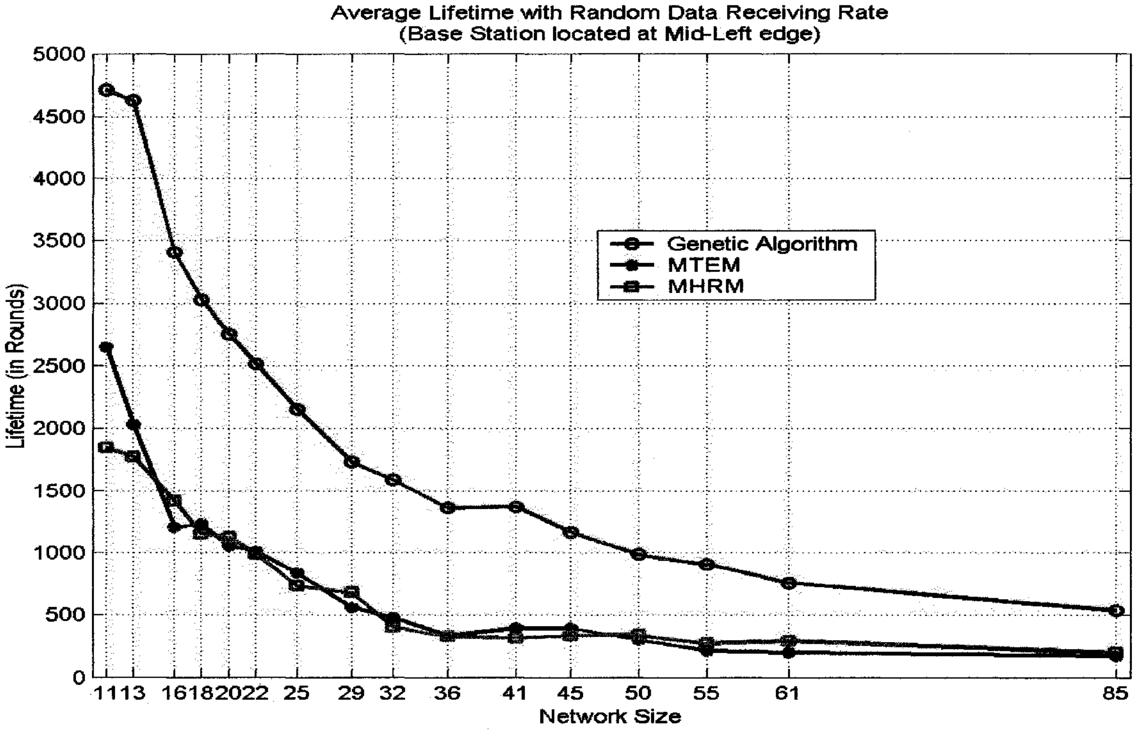


Figure 4.10 : Comparison of network lifetimes – Random data receiving rate (Base Station is located at mid-left edge)

Table 4.4 on the next page shows the 95% Confidence Interval (C.I.) chart with Upper Bound and Lower Bound of network lifetime (in *Rounds*) for our second set of experiments. This is interpreted in exactly the same way in Table 4.3, for the first set of experiments.

Network Size	Network Lifetime (in *Round*) with 95% Confidence Interval					
	Lifetime (in *Rounds*) obtained in GA		Percentage Improvement over Minimum Hop Routing Transmission (MHRM)		Percentage Improvement over Minimum Transmission Energy Model (MTEM)	
	Lower Bound	Upper Bound	Lower Bound	Upper Bound	Lower Bound	Upper Bound
11	4,144	5,277	199	349	148	226
13	4,366	4,886	249	273	204	260
16	3,165	3,652	212	281	239	348
18	2,859	3,197	241	288	209	302
20	2,546	2,960	211	291	214	335
22	2,282	2,748	222	294	207	311
25	1,959	2,347	236	373	233	288
29	1,639	1,813	234	284	255	396
32	1,452	1,718	350	448	281	403
36	1,209	1,509	333	573	367	460
41	1,276	1,463	384	510	292	469
45	1,094	1,232	283	450	275	329
50	905	1,060	259	332	271	439
55	830	972	287	398	364	511
61	661	848	201	340	317	468
85	500	567	225	356	282	363

Table 4.4 : Statistical Analysis of Improvement of Network Lifetime using GA (when Base Station at Mid-Left Edge of Network Area)

4.4 Varying the Path Loss Factor

In wireless communications, path loss normally includes the *propagation losses* caused by the environment and can be represented by the path loss exponent (*m*). The value of *m* is normally in the range of 2 to 4, where *m*=2 for the propagation in free space and *m*=4 for relatively lossy environments [Pan03]. In some environments, such as buildings, stadiums and other indoor environments, the value of *m* varies from 4 to 6. In our next experiment, we have considered the value of *m* as 2, 3, 4 and 6 with the network area grid

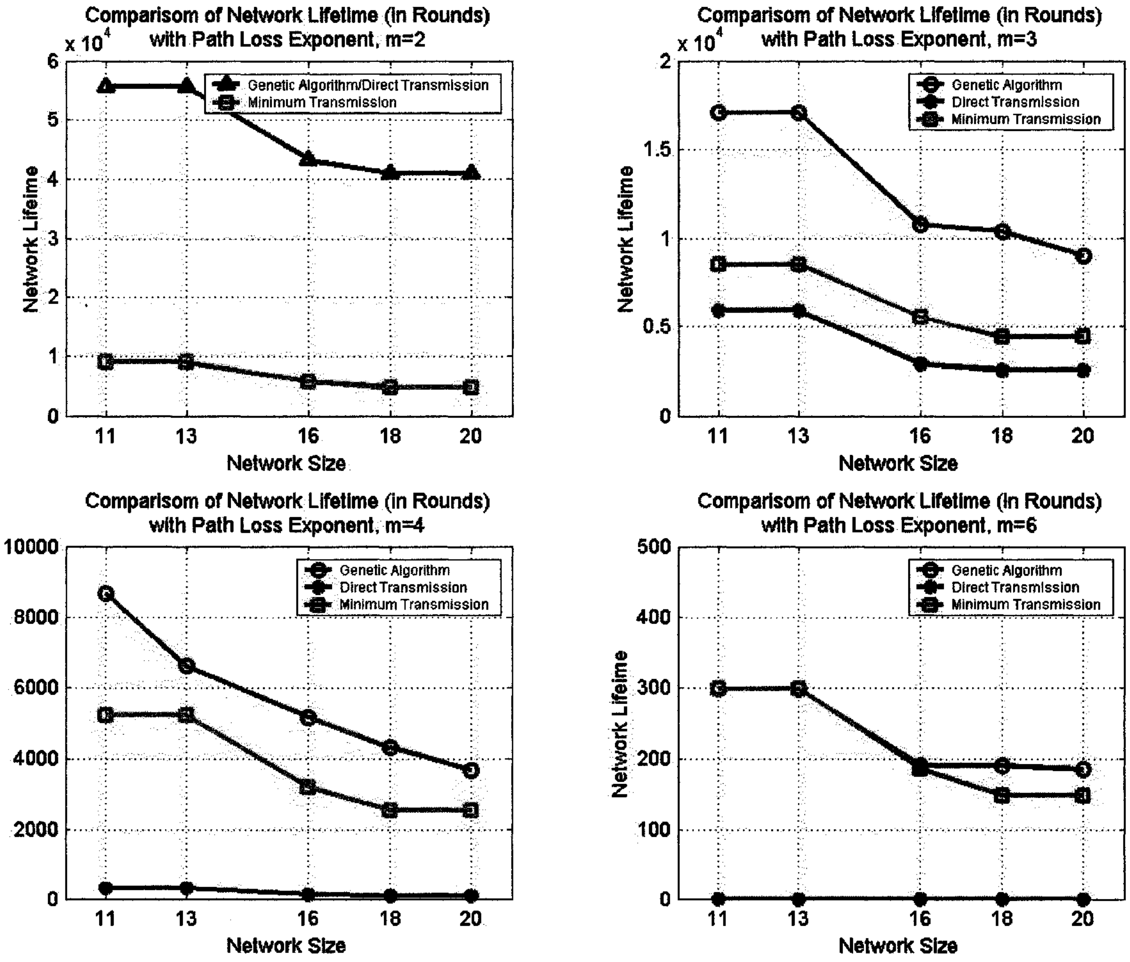


Figure 4.11 : Comparison of network lifetimes for the different Path Loss Exponents

size 4m x 4m. The simulation was performed over smaller size of networks – up to a 20 node network. The base station is placed at the corner of network area, and each relay node is considered to have a uniform data receiving rate (UDRR) of 1,000 bits/round. The results are shown in Fig. 4.11. It is observed that GA and Direct Transmission give the same lifetimes when m=2, which is in average more than 740% improvement over Minimum Transmission Energy Model. For $m = 3$, the GA lifetimes are found an average more than 250% improved over MHRM and MTEM. The network lifetimes are found almost nil for the Direct Transmission Model (MHRM) for higher values of m. Considering the path loss exponent value of 4, GA shows in average of 150% improvement over MTEM. Also, GA shows the same network lifetimes as MTEM when m=6 for the network sizes 11, 13 and 16.

4.5 Results for Very Large Networks

For our final set of experiments, we consider a sensing area of 480m x 480m (same as 85 nodes network, Table 4.1) but with 312 relay nodes, and with the base station located at the (0,0) coordinate with imaginary grid size of 20m x 20m . The size of the population is reduced to 100 as the number of nodes is very large. The data receiving rate is randomly chosen between 500 to 1,500 *bits/round* for each relay node from the sensor nodes of their corresponding cluster. The GA runs for a maximum of 300 generations, even though the maximum value of network lifetime is found in 143 generations. The GA achieves a network lifetime of 285 *rounds*. This represents an improvement of 149% and 155% compared to MHRM and MTEM respectively.

For this experiment, we set the other GA parameters as follows –

- Crossover type : Random
- Rate of Crossover : 1.0
- Rate of Mutation : 1.0
- Probability of Elitism : 0.8 and
- Elite organisms : 10%

The time required to process one generation is found 223 *msec* for the population size of 100, whereas for 85 nodes network each generation required 3.54 *sec* of processing time, with a population size of 10,000.

CHAPTER 5

Conclusions and Future Work

The lifetime of a wireless sensor network is typically constrained by the limited battery power of the network nodes. Therefore, it is extremely important to develop efficient, energy-aware routing protocols. Such protocols take into account the available energy of the network nodes and determine a routing schedule that attempts to maximize the network lifetime. ILP formulations for energy-aware routing can guarantee optimal solutions, but quickly become computationally intractable as the network size increases. For large networks, heuristic approaches, such as the minimum transmission energy model (MTEM) or the minimum hop routing model (MHRM), are typically used. Although such heuristics can quickly generate feasible solutions, the quality of the solutions is usually much lower, compared to the optimal solutions.

In this thesis, we present a genetic algorithm (GA) based method to find an energy-aware routing schedule, which tries to maximize the network lifetime. The network lifetime was measured in terms of the number of rounds of data transmission, before the first node runs out of power. We considered a range of network sizes, from 11 nodes to 85 nodes. We also considered different base station locations, as well as both fixed and variable data receiving rates for the network nodes. Another special experiment is carried out taking huge numbers of relay nodes within a dense imaginary grid. For each network, we compared the achieved lifetimes with optimal solutions (if an optimal solution could be generated) and heuristic routing schemes, such as MTEM and MHRM.

The GA approach was found to be very effective in generating good quality solutions in an efficient manner. Simulation results demonstrated that our GA approach was able to find the optimal solutions for small networks (up to 18 nodes). For larger networks, it was not possible to generate optimal solutions, so we compared our approach to MTEM and MHRM. The proposed GA approach was shown to improve the network lifetime by almost 200 to 300% compared to the traditional routing schemes - MTEM and MHRM. Finally, we repeated our experiments with different values of the path loss exponents *m* and found the performance of the GA to be as good as or better than the traditional routing schemes in all cases.

In this thesis, we assumed that the network nodes are stationary after deployment. In the future, we will investigate ways to augment our algorithm to take into account node mobility in each round. This work did not consider the bandwidth limitations of the wireless links. This can be taken into account in future versions of the algorithm.

References

1. [Aky02], I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci. Wireless sensor networks : a survey. *Computer Networks*, Vol 38, pages 393-422, 2002.
2. [AlK05], J.N. Al-Karaki and A.E. Kamal A. E. Routing techniques in wireless sensor networks: A survey. To be appeared in *IEEE Wireless Communications*, http://vulcan.ee.iastate.edu/~ kamal/Docs/kk04.pdf, 2005.
3. [Bari06a], A. Bari, A. Jaekel and S. Bandyopadhyay. Maximizing the Lifetime of Two-Tiered Sensor Networks. To appear in the *proceedings of the 6th IEEE International Conference on Electro/Information Technology*, MI USA, May 2006.
4. [Bari06b], A. Bari, A. Jaekel and S. Bandyopadhyay. Placement and Routing of Relay Nodes in Two-Tiered Sensor Networks, In *the proceedings of International Symposium on Broadband Access Technologies in Metropolitan Area Networks* (ISBAT2006), Niagara Falls, October 2006.
5. [Bea93], David Beasley, David R. Bull and Ralph R. Martin. *An overview of genetic algorithms*: Part 1, fundamentals, University Computing, Vol 15(2), pages 58-69, 1993.
6. [Bor05], M. Borghini, F. Cuomo, T. Melodia, U. Monaco and F. Ricciato. Optimal data delivery in wireless sensor networks in the energy and latency domains. In proceedings of the *First International Conference on Wireless Internet* (WICON'05), pages 138–145, 2005.
7. [Bri81], A. Brindle. *Genetic algorithms for function optimization.* PhD thesis, University of Alberta, 1981.
8. [Cam03], H. Cam, S. Ozdemir, D. Muthuavinashiappan and P. Nair. Energy efficient security protocol for wireless sensor networks. *Vehicular Technology Conference*, Oct 2003, Vol. 5, pages 2981-2984, 2003.
9. [Dav91], L. Davis. *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.
10. [Gaz99], C. Gazen and C. Ersoy. Genetic algorithms for designing multihop lightwave network topologies, Artificial Intelligence in Engineering, Vol. 13, pages 211–221, 1999.

11. [Gol89], D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, MA, 1989.
12. [Gol91], D.E. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In G.J.E. Rawlins, editor, *Foundations of Genetic Algorithms*, pages 69-93, Morgan Kaufmann, 1991.
13. [Gup03], G. Gupta and M. Younis. Performance evaluation of load-balanced clustering of wireless sensor networks. In *10th International Conference on Telecommunications*, Vol. 2, pages 1577–1583, 2003.
14. [Hei00], W. Heinzelman, A. Chandrakasan and H. Balakrishnan. Energy-Efficient Communication Protocol for Wireless Sensor Networks. In the *Proceeding of the 33rd Hawaii International Conference System Sciences (HICSS)*, Hawaii, pages 3005-3014, 2000.
15. [HTh00], W. Heinzelman. Application-specific Protocol Architectures for Wireless Networks. PhD thesis, Massachusetts Institute of Technology, 2000.
16. [Hil02], J. Hill and D. Culler. A wireless embedded sensor architecture for system-level optimization. Submitted for publication, 2002.
17. [Hol75], J. Holland. Adaptation in natural and artificial systems, University of Michigan Press, 1975.
18. [Hou05], Y.T. Hou, Y. Shi, J. Pan, S.F. Midkiff and K. Sohraby. On Energy Provisioning and Relay Node Placement for Wireless Sensor Networks. In *IEEE International Conference on Sensor and Ad Hoc Communications and Network.*
19. [Hou04], Y. T. Hou, Y. Shi, J. Pan, and S. F. Midkiff. Lifetime-optimal data routing in wireless sensor networks without flow splitting. In *Workshop on Broadband Advanced Sensor Networks*, San Jose, CA, 2004.
20. [Cplex91], ILOG CPLEX 9.1. Documentation. Available at the website http://www. columbia.edu/~dano/resources/cplex91_man/index.html
21. [Kah99], M.J. Kahn., R.H. Katz and K.S. Pister. Next century challenges: mobile networking for smart dust. In the *proceedings of ACM/IEEE MobiCom'99 Conferance*, pages 271-278, 1999.
22. [Kal02], K. Kalpakis, K. Dasgupta, and P. Namjoshi. Maximum Lifetime Data Gathering and Aggregation in Wireless Sensor Networks. *In the Proceedings of IEEE International Conference on Networking*, 2002.

23. [Liu02], Z. Liu, A. Jaekel and S. Bandyopadhyay. A Genetic Algorithm for Optimization of Logical Topologies in Optical Networks. Appears in *International Parallel and Distributed Processing Symposium (IPDPS),* pages 202-209, 2002.
24. [Lin02], S. Lindsey and C.S. Raghavendra. PEGASIS: Power Efficient GAthering in Sensor Information Systems. In the *Proceedings of the IEEE Aerospace Conference,* Montana, Vol 3, pages 1125-1130, 2002.
25. [Lof06], B. Loft and J. Snow. A genetic algorithm for drawing ordered sets. Texas College Mathematics Journal, Vol 3 (2), pages 10-26, 2006.
26. [Man02], A. Manjeshwar and D.P. Agarwal. APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks. In the *Proceedings of the International Parallel and Distributed Processing Symposium (IPDPS)*, pages 195-202, 2002.
27. [Mha04], V. Mhatre, and C. Rosenberg. Homogeneous vs. Heterogeneous Clustered Sensor Networks: A Comparative Study. In Proc. of *2004 IEEE International Conference on Communications (ICC 2004)*, France, pages 856-861, 2004.
28. [Pan05], J. Pan, L. Cai, Y.T. Hou, Y. Shi and S.X. Shen. Optimal Base-Station Locations in Two-Tiered Wireless Sensor Networks. *IEEE Transaction on Mobile Computing*, Vol. 4 (5), pages 458-473, 2005.
29. [Pan03], J. Pan, Y.T. Hou, I. Cai, Y. Shi and S.X. Shen. Topology Control for Wireless Sensor Networks. In Proc. Of *Ninth Annual International Conference on Mobile Computing and Networking*, pages 286-299, 2003.
30. [Rag02], V. Raghunathan, C. Schurgers, S. Park and M.B. Srivastava. Energy-aware wireless microsensor networks. *IEEE Signal Processing Magazine*, Vol. 19(2), pages 40-50, 2002.
31. [Rec73], I. Rechenberg. Evolutionsstrategie: Optimierung technischer System nach Prinzipien der biologischen evolution, Frommann-Holzboog, Stuttgart, 1973.
32. [Sas05], K. Sastry, D. Goldberg and G. Kendall. *Genetic Algorithms*. Chapter 4 of Introductory Tutorials in Optimization and Decision Support Techniques, (eds. E. Burke and G. Kendall), pages 97-125, Kluwer, 2005.
33. [Shu05], T. Shu, M. Krunz and S. Vrudhula. Power Balanced Coverage-Time Optimization for Clustered Wireless Sensor Networks. In *Proceedings of ACM International Symposium on MobiHoc'05*, USA, pages 111-120, 2005.

34. [Spe94], W.M. Spears and K.A. De Jong. 1994, On the virtues of parameterized uniform Crossover. In *Proceedings of the 4th International Conference on Genetic Algorithms*, 1994.
35. [Sys89], G. Syswerda. Uniform crossover in genetic algorithms. In *Proceedings of the 3rd International Conference on Genetic Algorithms,* pages 2–9, 1989.
36. [Tan06], J . Tang, B. Hao, and A. Sen. Relay node placement in large scale wireless sensor networks. *Computer Communications*, Vol. 29(4), pages 490–501, 2006.
37. [Yar05], M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu and S. Singh. Exploiting heterogeneity in sensor networks. In proceedings of INFOCOM 2005, *24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Vol. 2, pages 878–890, 2005.
38. [Yao02], Y. Yao and J. Gehrke. Query Processing for Sensor Networks. In proceeding of *Conference on Innovation Data System Research (CIDR)*, 2003.
39. [Yon03], M. Younis, M. Youssef and K. Arisha. Energy-Aware Management for Cluster-Based Sensor Networks. *Journal of Computer Networks*, Vol 43 (5), pages 649-668, 2003.
40. [Zha05], M. Zhang, J. Song and Y. Zhang. Three-Tiered Sensor Networks Architecture for Traffic Information Monitoring and Processing. Appears in *Intelligent Robots and Systems (IROS'05)*, pages 2291-2296, 2005.

Appendix A : Definitions

Allele - The value of each element(Gene) in a chromosome is known as allele

Base Station – The end node in a network that collects all data from sensor nodes either directly or via one or more relay nodes

Chromosome – An abstract, coded representation of a solution to the target problem

Cluster Head – The node which receives data from corresponding sensor nodes and transmits to the base station or any other cluster head

Convergence – A situation where individuals in a population become identical or similar to each other, which indicates that GA has slowed to a point that it does not seem to find new, better solution

CPLEX – An optimization software package to solve integer programming (IP) as well as integer linear programming (ILP)

Critical Node – The node which runs out of its battery power first

Crossover - A genetic operator which creates new chromosomes by combining genetic materials from the two parent chromosomes

Crossover Point – The split position in crossover. There might be more than one crossover points depending on the type of crossover

Crossover Probability – Percentage of individuals in a population that can perform the crossover process

Destination Node – The node that collects data from other node(s)

Elitism – The replacement process where the poorest individuals of the current generation is replaced by appending same number of the best performing individuals from the previous generation

Elite Organisms – Percentage of individuals of a population that replace the same number of individuals from the current generation

Elite Probability – Percentage of generations that perform the Elitism process

Fitness – A value assigned to an individual which indicates the performance of this individual for the target problem. It also represents the possibility of being selected for reproduction

Fitness Function – The function that maps an individual to a fitness value

Flow-splitting Routing – The routing in which data transmits from a single node towards multiple destinations simultaneously

Gene - Each element in a chromosome is called gene which contains a part of solution

Generation - The population generated by each reproduction

Hierarchical Routing – The routing which performs on cluster based networks, cluster-heads collects data from the respective cluster and forwards them towards the base station

Individual – A single chromosome in a population

Multi-hop Routing – When the cluster-heads transmits data towards the base station via one or more other cluster-heads

Mutation - A genetic operator which creates a new chromosome by making random alternation to one or more values of the genes in a chromosome

Mutation Probability – Percentage of individuals in a population that can perform the mutation process

Non flow-splitting Routing – In this routing data transmits from a node towards a single destination node

Offspring – The new chromosome generated by the reproduction or crossover process

Population – A group of chromosomes which mate with each other during crossover to produce new offspring

Relay Node – Nodes with special functionalities deployed in hierarchical WSN

Round – Each period of data gathering by the replay node from corresponding sensor nodes and transmitting those to base station

Selection - The mechanism that according to their fitness, two chromosomes are selected for mutation

Single-hop Routing – When the cluster-heads send data to the base station directly provided that the base station lies with the transmission range of all cluster-heads

Source Node – The node that transmit data to other node(s)

Stopping Criteria – The conditions under which a GA should stop producing any new generation

Appendix B : Sample Data Sheet

0	80	160	56.5685	126.4911	203.9608	80	113.1371	178.8854	126.4911	169.7056	233.2381	160	178.8854	226.2742	256.125
80	0	80	56.5685	56.5685	126.4911	113.1371	80	113.1371	126.4911	126.4911	169.7056	178.8854	160	178.8854	200
160	80	0	126.4911	56.5685	56.5685	178.8854	113.1371	80	169.7056	126.4911	126.4911	226.2742	178.8854	160	164.9242
56.5685	56.5685	126.4911	0	80	160	56.5685	56.5685	126.4911	80	113.1371	178.8854	126.4911	126.4911	169.7056	200
126.4911	56.5685	56.5685	80	0	80	126.4911	56.5685	56.5685	113.1371	80	113.1371	169.7056	126.4911	126.4911	144.2221
203.9608	126.4911	56.5685	160	80	0	203.9608	126.4911	56.5685	178.8854	113.1371	80	233.2381	169.7056	126.4911	120
80	113.1371	178.8854	56.5685	126.4911	203.9608	0	80	160	56.5685	126.4911	203.9608	80	113.1371	178.8854	215.4066
113.1371	80	113.1371	56.5685	56.5685	126.4911	80	0	80	56.5685	56.5685	126.4911	113.1371	80	113.1371	144.2221
178.8854	113.1371	80	126.4911	56.5685	56.5685	160	80	0	126.4911	56.5685	56.5685	178.8854	113.1371	80	89.4427
126.4911	126.4911	169.7056	80	113.1371	178.8854	56.5685	56.5685	126.4911	0	80	160	56.5685	56.5685	126.4911	164.9242
169.7056	126.4911	126.4911	113.1371	80	113.1371	126.4911	56.5685	56.5685	80	0	80	126.4911	56.5685	56.5685	89.4427
233.2381	169.7056	126.4911	178.8854	113.1371	80	203.9608	126.4911	56.5685	160	80	0	203.9608	126.4911	56.5685	40
160	178.8854	226.2742	126.4911	169.7056	233.2381	80	113.1371	178.8854	56.5685	126.4911	203.9608	0	80	160	200
178.8854	160	178.8854	126.4911	126.4911	169.7056	113.1371	80	113.1371	56.5685	56.5685	126.4911	80	0	80	120
226.2742	178.8854	160	169.7056	126.4911	126.4911	178.8854	113.1371	80	126.4911	56.5685	56.5685	160	80	0	40
256.125	200	164.9242	200	144.2221	120	215.4066	144.2221	89.4427	164.9242	89.4427	40	200	120	40	0

Distance Matrix for 16 Node Network
with grid area 40m x 40m and Base Station at Corner

0	80	160	56.5685	126.4911	203.9608	80	113.1371	178.8854	215.4066	126.4911	169.7056	233.2381	160	178.8854	226.2742
80	0	80	56.5685	56.5685	126.4911	113.1371	80	113.1371	144.2221	126.4911	126.4911	169.7056	178.8854	160	178.8854
160	80	0	126.4911	56.5685	56.5685	178.8854	113.1371	80	89.4427	169.7056	126.4911	126.4911	226.2742	178.8854	160
56.5685	56.5685	126.4911	0	80	160	56.5685	56.5685	126.4911	164.9242	80	113.1371	178.8854	126.4911	126.4911	169.7056
126.4911	56.5685	56.5685	80	0	80	126.4911	56.5685	56.5685	89.4427	113.1371	80	113.1371	169.7056	126.4911	126.4911
203.9608	126.4911	56.5685	160	80	0	203.9608	126.4911	56.5685	40	178.8854	113.1371	80	233.2381	169.7056	126.4911
80	113.1371	178.8854	56.5685	126.4911	203.9608	0	80	160	200	56.5685	126.4911	203.9608	80	113.1371	178.8854
113.1371	80	113.1371	56.5685	56.5685	126.4911	80	0	80	120	56.5685	56.5685	126.4911	113.1371	80	113.1371
178.8854	113.1371	80	126.4911	56.5685	56.5685	160	80	0	40	126.4911	56.5685	56.5685	178.8854	113.1371	80
215.4066	144.2221	89.4427	164.9242	89.4427	40	200	120	40	0	164.9242	89.4427	40	215.4066	144.2221	89.4427
126.4911	126.4911	169.7056	80	113.1371	178.8854	56.5685	56.5685	126.4911	164.9242	0	80	160	56.5685	56.5685	126.4911
169.7056	126.4911	126.4911	113.1371	80	113.1371	126.4911	56.5685	56.5685	89.4427	80	0	80	126.4911	56.5685	56.5685
233.2381	169.7056	126.4911	178.8854	113.1371	80	203.9608	126.4911	56.5685	40	160	80	0	203.9608	126.4911	56.5685
160	178.8854	226.2742	126.4911	169.7056	233.2381	80	113.1371	178.8854	215.4066	56.5685	126.4911	203.9608	0	80	160
178.8854	160	178.8854	126.4911	126.4911	169.7056	113.1371	80	113.1371	144.2221	56.5685	56.5685	126.4911	80	0	80
226.2742	178.8854	160	169.7056	126.4911	126.4911	178.8854	113.1371	80	89.4427	126.4911	56.5685	56.5685	160	80	0

Distance Matrix for 16 Node Network
with grid area 40m x 40m and Base Station at Mid-Left Boundary

0	8	16	5.6569	12.6491	20.3961	8	11.3137	17.8885	12.6491	16.9706	23.3238	16	17.8885	22.6274	25.6125
8	0	8	5.6569	5.6569	12.6491	11.3137	8	11.3137	12.6491	12.6491	16.9706	17.8885	16	17.8885	20
16	8	0	12.6491	5.6569	5.6569	17.8885	11.3137	8	16.9706	12.6491	12.6491	22.6274	17.8885	16	16.4924
5.6569	5.6569	12.6491	0	8	16	5.6569	5.6569	12.6491	8	11.3137	17.8885	12.6491	12.6491	16.9706	20
12.6491	5.6569	5.6569	8	0	8	12.6491	5.6569	5.6569	11.3137	8	11.3137	16.9706	12.6491	12.6491	14.4222
20.3961	12.6491	5.6569	16	8	0	20.3961	12.6491	5.6569	17.8885	11.3137	8	23.3238	16.9706	12.6491	12
8	11.3137	17.8885	5.6569	12.6491	20.3961	0	8	16	5.6569	12.6491	20.3961	8	11.3137	17.8885	21.5407
11.3137	8	11.3137	5.6569	5.6569	12.6491	8	0	8	5.6569	5.6569	12.6491	11.3137	8	11.3137	14.4222
17.8885	11.3137	8	12.6491	5.6569	5.6569	16	8	0	12.6491	5.6569	5.6569	17.8885	11.3137	8	8.9443
12.6491	12.6491	16.9706	8	11.3137	17.8885	5.6569	5.6569	12.6491	0	8	16	5.6569	5.6569	12.6491	16.4924
16.9706	12.6491	12.6491	11.3137	8	11.3137	12.6491	5.6569	5.6569	8	0	8	12.6491	5.6569	5.6569	8.9443
23.3238	16.9706	12.6491	17.8885	11.3137	8	20.3961	12.6491	5.6569	16	8	0	20.3961	12.6491	5.6569	4
16	17.8885	22.6274	12.6491	16.9706	23.3238	8	11.3137	17.8885	5.6569	12.6491	20.3961	0	8	16	20
17.8885	16	17.8885	12.6491	12.6491	16.9706	11.3137	8	11.3137	5.6569	5.6569	12.6491	8	0	8	12
22.6274	17.8885	16	16.9706	12.6491	12.6491	17.8885	11.3137	8	12.6491	5.6569	5.6569	16	8	0	4
25.6125	20	16.4924	20	14.4222	12	21.5407	14.4222	8.9443	16.4924	8.9443	4	20	12	4	0

Distance Matrix for 16 Node Network
with grid area 4m x 4m and Base Station at Corner

Some sample results obtained from network of sizes 11 to 25, where
Data receiving rate : UDRR
Base station location : Corner of network area
Grid area : 40m x 40m

A. Genetic Algorithm (GA)

Network Size	Final Gene value	Network Lifetime	Critical Node#
11	2 5 8 7 8 4 10 11 10 11	4000	8
13	2 7 8 10 13 12 10 9 12 13 13 13	3355	5
16	10 9 9 3 12 16 11 12 12 11 15 16 14 15 16	2824	12
18	6 6 14 12 13 10 18 6 14 14 16 13 17 18 13 17 18	2440	10
20	8 9 14 8 16 4 15 12 16 15 20 16 17 15 19 20 18 19 20	2183	16
22	9 10 4 11 16 17 17 9 10 14 18 16 22 18 19 17 21 22 20 21 22	1960	18
25	6 5 17 5 12 14 21 13 17 25 9 16 17 21 23 20 21 23 25 24 25 24 24 25	1780	21
………			
………			

B. Minimum Hop Routing Model (MHRM)

Network Size	Final Gene value	Network Lifetime	Critical Node#
11	11 11 11 11 11 11 11 11 11 11	1234	1
13	13 13 13 13 13 13 13 13 13 13 13 13	1234	1
16	9 16 16 16 16 16 15 16 16 16 16 16 16 16 16	1234	2
18	13 18 18 13 18 18 18 17 18 18 13 18 18 18 18 18 18	1234	2
20	15 16 12 20 20 15 20 20 20 19 20 20 15 20 20 20 20 20 20	1234	4
22	10 17 18 22 14 22 22 17 22 22 22 21 22 22 17 22 22 22 22 22 22	1160	10
25	16 17 17 13 20 21 25 17 25 25 20 25 25 25 24 25 25 20 25 25 25 25 25 25	1160	13
………			
………			

C. Minimum Transmission Energy Model (MTEM)

Network Size	Final Gene value	Network Lifetime	Critical Node#
11	4 5 5 7 8 9 10 11 7 11	2439	7
13	4 5 6 7 7 9 10 11 12 13 9 13	2439	7
16	4 5 6 8 9 9 10 11 12 14 15 16 10 11 16	1510	11
18	5 6 7 8 9 10 10 12 13 14 15 16 17 18 12 13 18	1205	13
20	4 5 7 8 9 10 11 12 12 14 15 16 17 18 19 20 14 15 20	1204	15
22	1 6 7 1 2 3 4 5 6 7 9 9 10 11 12 13 14 16 16 17 18	1094	6
25	5 6 7 8 9 10 10 12 13 14 15 16 17 17 19 20 21 22 23 24 25 19 20 25	1094	17
………			
………			

Vita Auctoris

Name : Md. Shamsul Wazed
Place of Birth : Dhaka, Bangladesh
Year of Birth : 1967

Education : Notre Dame College
Dhaka, Bangladesh, 1984

Bachelor of Science in Civil Engineering
Rajshahi University of Engineering and Technology (RUET)
Rajshahi, Bangladesh, 1991

Bachelor of Science in Computer Science
University of Windsor
Windsor, Ontario, Canada, 2004

Masters of Science in Computer Science
University of Windsor
Windsor, Ontario, Canada, 2007