Multi-Objective Optimization of Clustering Techniques in WSN using Harmony Search Algorithm

Submitted By

Riddhi Parsania

14mcen12

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Multi-Objective Optimization of Clustering Techniques in WSN using Harmony Search Algorithm

Major Project

Submitted in partial fulfillment of the requirements

for the degree of

Master of Technology in Computer Science and Engineering(Networking Technologies)

Submitted By

Riddhi Parsania

14mcen12

Guided By

Dr.Gaurang Raval

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF TECHNOLOGY

NIRMA UNIVERSITY

AHMEDABAD-382481

May 2016

Certificate

This is to certify that the major project entitled **"Multi-Objective Optimization of Clustering Techniques in WSN using Harmony Search Algorithm"** submitted by **Riddhi Parsania (Roll No: 14MCEN12)**, towards the partial fulfillment of the requirements for the award of degree of Master of Technology in Networking Technologies(CSE) of Nirma University, Ahmedabad, is the record of work carried out by him under my supervision and guidance. In my opinion, the submitted work has reached a level required for being accepted for examination. The results embodied in this major project part-II, to the best of my knowledge, haven't been submitted to any other university or institution for award of any degree or diploma.

Dr. Gaurang Raval
Guide & Associate Professor,
Coordinator M.Tech-CSE(NT),
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr. Sanjay Garg
Professor and Head,
CSE Department,
Institute of Technology,
Nirma University, Ahmedabad.

Dr P. N. Tekwani
Director,
Institute of Technology,
Nirma University, Ahmedabad

Statement of Originality

I, **Riddhi Parsania**, Roll. No. **14MCEN12**, give undertaking that the Major Project entitled "**Multi-Objective Optimization of Clustering Techniques in WSN using Harmony Search Algorithm**" submitted by me, towards the partial fulfillment of the requirements for the degree of Master of Technology in **Computer Science & Engineering(Networking Technologies)** of Institute of Technology, Nirma University, Ahmedabad, contains no material that has been awarded for any degree or diploma in any university or school in any territory to the best of my knowledge. It is the original work carried out by me and I give assurance that no attempt of plagiarism has been made.It contains no material that is previously published or written, except where reference has been made. I understand that in the event of any similarity found subsequently with any published work or any dissertation work elsewhere; it will result in severe disciplinary action.

_____

Signature of Student

Date:

Place:

Endorsed by

Dr. Gaurang Raval

Acknowledgements

It gives me immense pleasure in expressing thanks and profound gratitude to **Dr. Gaurang Raval**, Associate Professor, Computer Science Department, Institute of Technology, Nirma University, Ahmedabad for his valuable guidance and continual encouragement throughout this work. The appreciation and continual support he has imparted has been a great motivation to me in reaching a higher goal. His guidance has triggered and nourished my intellectual maturity that I will benefit from, for a long time to come.

It gives me an immense pleasure to thank **Dr. Sanjay Garg**, Hon'ble Head of Computer Science and Engineering Department, Institute of Technology, Nirma University, Ahmedabad for his kind support and providing basic infrastructure and healthy research environment.

A special thank you is expressed wholeheartedly to **Dr P.N.Tekwani**, Hon'ble Director, Institute of Technology, Nirma University, Ahmedabad for the unmentionable motivation he has extended throughout course of this work.

I would also thank the Institution, all faculty members of Computer Engineering Department, Nirma University, Ahmedabad for their special attention and suggestions towards the project work.

- Riddhi Parsania

14MCEN12

Abstract

Now a days we all know that wireless sensor network (WSN) is used in many application like commercial, military, and environmental. Clustering is one of the popular method. Energy consumption optimization is a major focus while planning and the designing the operation of wireless sensor network. The lifetime of a network can be extended by clustering technique which includes data aggregation and balancing the energy consumption among the sensor nodes. The algorithms which are generally ussed are heuristic and and they aim to generate minimum number of cluster in a WSN and also to keep minimum distance between them.So here we are using some protocol like LEACH(low energy adaptive clustering hierarchy), GCA(genetic clustering algorithm), Fuzzy-C mean, Particle Swarm Optimization (PSO), ERP(Evolutionary Routing Protocol), EAERP(Energy Aware Evolutionary Routing Protocol), HSA(Harmony Search Algorithm) to know that which protocol is best for energy consumption, which is minimize the transmission distance etc. Heuristic methods suggest probable solutions from which best objective function is selected as maximum or minimum valued function. Depending on the optimization function design the choice may be to have minimum value or maximum value. There can be sub-objectives also, which may be conflicting in nature. The multi-objective function can not have unique solutions So there is a possibility of multiple probable solutions. The same has been explore in this thesis.

Key words : Clustering, Network lifetime, Energy consumption, Wireless sensor network, LEACH, Harmony search algorithm(HSA), multi-objective optimization.

_

Contents

List of Figures

Chapter 1

Introduction

1.1 Wireless Sensor Network

Wireless Sensor networks are composed of large number of sensor nodes with batteries having low capacity. Wireless networks of thousands of inexpensive tiny devices are capable of communication, computation and sensing. A sensor node has limited capability of sensing, computing, storing and transmitting data. Wireless Sensor networks provide a bridge between the real physical and virtual worlds. In the wireless sensor networks (WSNs), the sensor nodes are dropped in a the area for monitoring purpose. The nodes in these networks coordinate to produce high-quality information and each of these scattered sensor nodes has the capabilities to collect and route data back to the base stations, which are fixed or mobile. Figure 1.1 shows each node in a sensor network is typically equipped with one or more sensors, a radio transceiver or other wireless communications device, a small micro-controller, and an energy source.

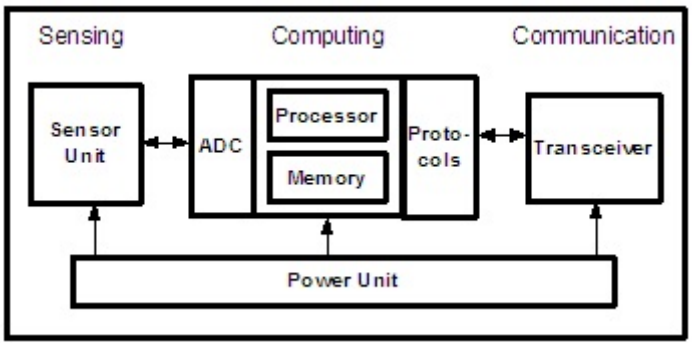

Figure 1.1: Architecture of sensor node

In WSNs, conservation of energy, which is directly related to network lifetime, is considered relatively more important than the performance of the network in terms of reliability of data sent. Energy conservation plays an important role in wireless sensor network as nodes are having limited capability in this context. While developing a routing protocol for wireless sensor networks, energy usage is considered to be the most important factor.

Routing in WSNs has several generic types like flat routing, hierarchical routing, location based routing etc. In flat routing every node perform the sensing task and does similar job. In this technique, data transmission is done in hop by hop manner. Hierarchical Routing eliminates the delay involved which is present in hop by hop transmission. In hierarchical architecture, cluster formation has been done and each cluster contains one cluster head with high energy and others are cluster members. Higher energy nodes can be used to process and send the information, while low-energy nodes can be used to perform the sensing job. Many routing protocols have been proposed in the literature such as LEACH[1][2], K-Means[2], ERP(Evolutionary Routing Protoco)[3], EAERP(Energy Aware Evolutionary Routing Protocol)[3], FCM(Fussy-C Mean)[2], HSA(Harmony Search Algorithm)[2][4].

There are large number of arithmetic operations required to transfer one bit of data to the BS. The physical environment produce very similar data in nearby nodes and transmitting data redundantly to BS. Therefore, clustering of sensor nodes such that data from sensor nodes of a group can be combined or compressed together in an proper way and transmit only compact data to BS. This reduces the traffic and energy consumption of nodes. The process of grouping of sensor nodes in a densely deployed sensor network is known as clustering.

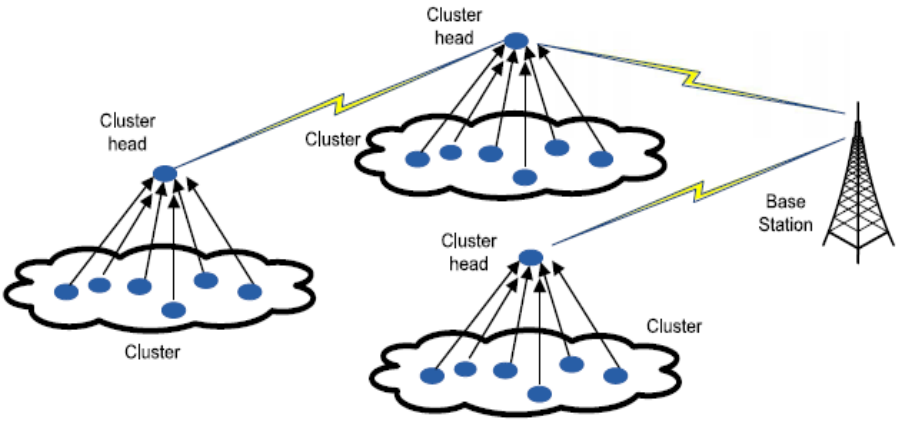

Figure 1.2: Clustering in WSN

1.2 Application of Wireless Sensor network

There are wide ranging applications of sensor network. It can vary significantly in application requirements, mode of deployment, sensing modality.

- Environmental monitoring (e.g. traffic, territory , security)
- Industrial and diagnostics
- Health care monitoring
- Surveillance system
- Target tracking
- Explosive & Nuclear things

1.3 Meta-Heuristics Protocols

Heuristic algorithms try to find a good solution for optimization problem in a some amount of computing time. There is no assurance to find the best or optimal solution, so it may find a better or improved solution than an good guess. It is widly excepted that heuristic methods are local search methods. Because their searches focus on the local variations, and the optimal or best solution can locate outside of this local region. However, a high-quality feasible solution in the local region of interest is usually accepted as

a good solution in many optimization problems in practice if time is the major constraint.

Two different kind of heuristics are identified. First is Constructive Heuristics and second is Search Heuristics (Heuristic Search Strategies). Constructive heuristics are mainly problem specific and try to construct one single solution with the best possible quality by taking care of selecting promising solution elements. Search heuristics implement a search in the solution space of a given problem during which they examine many different solutions in order to find the best possible one[5]. Three problem-independent, basic principles of heuristic search can be identified repeated solution construction where a new solution is obtained by constructing a new one from scratch, repeated solution modification where a new solution is obtained by modifying an existing one, and repeated solution recombination where a new solution is obtained by recombining two or more existing solutions[3].

The actual realization of the basic principle itself, however, is a problem specific issue, since it has to be defined how a solution may be constructed, modified or recombined. This distinction between problem independent and problem specific aspects of search heuristics reveals a major advantage: The search strategies and the problem specific parts can be implemented independently from each other[5].

Chapter 2

Literature Survey

2.1 LEACH Protocol

One most popular hierarchical routing protocol is low energy adaptive clustering hierarchy (LEACH)[6]. In this protocol network divides into cluster to gain energy and scalability. In the location-based routing protocol, information of nodes is used to compute the routing path in the network. In the LEACH network forms a clusters by using a assigning algorithm, where nodes will decide cluster head. Firstly a node will decide the probability of the Cluster Head and broadcast the conclusion. Each non CH nodes check its cluster by selecting the CH that can be get by using minimum energy in terms of communication. Then CHs balance the load in overall network. The load is then balanced by the Cluster Heads over the entire network. A random number T is selected anywhere between zero and one by each sensor. A node will take the position of a cluster head if the number obtained is lower than the threshold calculated.

$$T_{(s)} = \begin{cases} \frac{P}{1-P\times(r \bmod 1/p)} & : if\ S \epsilon G' \\ 0 & : Otherwise \end{cases} \tag{2.1}$$

Here, percentage of cluster head nodes is given by P, current round number is given by r, and G represents the nodes that have not been selected as cluster heads in the last 1/p rounds.

LEACH is homogeneous WSNs. Hierarchical routing is managed by stable election protocol(SEP)[3] in heterogeneous sensor networks. These advanced nodes create hetero-

geneity. In SEP, The weighting is performed to decide the election probabilities and it is dependent upon the initial energy of node relative to other nodes. Lets suppose that E0 is initial energy of each normal sensor. If the fraction of advance node is m and the energy fraction factor between advance node and normal node is then[1],

$$p_{nrm} = \frac{p}{(1 + \alpha \times m)} \tag{2.2}$$

$$p_{adv} = \frac{p}{(1 + \alpha \times m)} \times (1 + \alpha) \tag{2.3}$$

Then, in SEP the threshold value is re-placed by the normal node and advance node as follow:

$$T(S_{nrm}) = \begin{cases} \frac{P_{nrm}}{1 - P_{nrm} \times (r \bmod 1/P_{nrm})} : if\ S_{nrm}\ \epsilon\ G' \\ 0 \qquad\qquad\qquad\qquad\quad : Otherwise \end{cases} \tag{2.4}$$

$$T(S_{adv}) = \begin{cases} \frac{P_{adv}}{1 - P_{adv} \times (r \bmod 1/P_{adv})} : if\ S_{adv}\ \epsilon\ G" \\ 0 \qquad\qquad\qquad\qquad\quad : Otherwise \end{cases} \tag{2.5}$$

Here, r is the ongoing round. The nodes that havent been appointed as the cluster heads in the previous $1/P_{nrm}$ rounds are represented by G and $T(S_{nrm})$ is the threshold applied to a population of $n * (1 - m)$ normal nodes[3]. This will ensure that each node which is normal, will get a chance to be a cluster head only once every$1/P_{nrm} * (1 + \alpha * m)$ rounds per epoch, and that the average number of cluster heads that are normal nodes per round per epoch is equal to $n * (1 - m) * P_{nrm}$. In the same way, the nodes that are advanced and those that have not yet been appointed as the CH, are represented by G. $T(S_{adv})$ is the threshold applied to a population of n m advanced nodes. This is to ensure that every advanced nodegets a chance to become a CH once and only once every $(1/P) * (1 + \alpha * m)/(1 + \alpha)$[3].

2.2 Genetic Clustering Algorithm

As the name suggests, GCA is based on a randomized optimization and search technique trained by the fundamentals of genetics and natural evolution. This algorithm searches for a close, but not exactly optimal solution for the fitness function or the objective function for any optimizable problem. The parameters in such an algorithm are coded into strings, commonly called chromosomes. These strings, when grouped together become the population for this algorithm. Like in any clustering algorithm, first, a random population is constructed to show the different points in the search space. The functions, objective and fitness, is assigned to every string, which depicts the goodness degree of that string. Since this algorithm revolves around the principles of genetics, a few strings are chosen based on survival of the fittest. These strings are then given several copies, which are then put into a mating pool[3]. Operators like crossover and mutation, which are again biologically inferred, are applied to these strings, so that they generate a new set, or generation of strings. The process of selection, crossover, and mutation as described above are repeated either for a fixed amounts of generations, or until a terminating condition or case is met.

In the present research, population-based meta-heuristic cluster-based routing in WSN is carried out with a focus on evolutionary algorithms. In the GA, find proper CHs to least the total network distance. It applies binary individual. 1 represent that the sensor is a CH; any other way, it is a regular node.

$$\varphi_{GA}(I) = \omega * (TD - D) + (1 - \omega) * (TN - N_{CH}) \tag{2.6}$$

Where, TD stands for the summation of the distance between every node and BS. Sum of the distance between all regular nodes and CH is D. TN stands for total number of nodes. W, is the weight that is pre-defined. All parameters apart from D, and N(CH), are constant in the WSN topology. The near the D, or the lower the number of Node Cluster Head (NCH), the fitness value is higher. GA tries to find good fitness function which is minimized.

$$\varphi_{GCA}(I) = \omega * N_{CH} + (1 - \omega) * ND \tag{2.7}$$

GCA, also known as the Genetic Clustering Algorithm suggests for increasing network lifetime by cutting down on the dissipated energy. Every chromosome is represented using a list of fixed length, which amounts to the size of the network. The value 1 is used as the gene value to represent a dead node, or a number greater than 0 to the node number of its Cluster Head. The Genetic clustering Algorithms fitness function, allows for the optimization of 3 attributes. These are: The number of nodes that are Cluster Heads N(ch) Distance between every node within each cluster, to the cluster head ND, which is also known as the Euclidian distance. A weight (w), for adjusting both the parameters mentioned above, and to find each chromosomes fitness value. This weight is system-dependent.

2.3 Evolutionary Routing Protocol (ERP)

A standout amongst the most prominent meta-heuristics is Evolutionary Algorithms. EAs motivated by the force of characteristic development, have been broadly utilized as pursuit and advancement devices in different issue areas. The general EA system begins with a beginning populace of arrangements and obliges distinctive types of choice and variety administrators to create new arrangements, by using an iterative procedure. The greater the capability of an individual, the greater is the risk with which this has to be chosen. The nature of an distinct is computed by its fitness. There are numerous variety administrators to change data in the folks. On the off chance that quality trade is done between two or more people, the variety administrator is called parents[1]. In the event that the qualities of one individual changes all alone, the variety administrator is called mutation. Regularly, EA accepts that client customizable state under which the task of recombination and mutation can be easily performed. The entire procedure of choice and varieties constitutes one era or cycle of an EA, and the developing procedure can keep on advancing eras until a predefined end condition is fulfilled

$$Compactness = \sum_{i=1}^{CHs} \sum_{\forall n \epsilon C_i} d(n, CH_i) \tag{2.8}$$

Presently we speak to ERP with changed fitness capacity. The issue basic to an effective EA execution is the decision of a decent fitness capacity. It frames the premise for

determination and in this manner encourages enhancements. From the critical thinking point of view, it speaks to the assignment to be tackled in the developmental setting.Actually, this is a capacity that relegates a value compute to single arrangements. This is a motivating factor to reconsider the evolutionary clustering protocols decision to choose the fitness function. While EA based protocols beat LEACH in prolonging the lifetime, they disappoint to get better strength duration[1]. This is the essential element used to provide least system vitality utilization. Therefore, in order to be more accurate in deciding the reliability of the clustering answer that the routing protocol provides, intra-cluster distance can be quantified by above equation.

CH, as discussed above, showes the number of clustering heads. Ci is the ith cluster differentiate with cluster-head CHi, and any non cluster head node, n, lies to the cluster Ci for which, the smallest distance between n and Chi is satisfied. Further, quantification of the inter-distance as the least Euclidean distance between any set of cluster heads can also be carried out.

$$d_{min} = \min_{\forall C_i, C_j, C \neq C_j} d(CH_i, CH_j) \tag{2.9}$$

Then, the objective (i.e. fitness) function is to simultaneously minimize f1= Compactness/dmin and f2=number of CH. Accordingly, where w is a predefined weight.

$$Fitness(Chrom_{ERP}) = \omega \times f_1 + (1 - \omega) \times f_2 \tag{2.10}$$

2.4 Energy Aware Evolutionary Routing Protocol (EAERP)

For EAERP, the suggested objective function is to minimize the total sum of the dissipated energy which can be computed as the energy dissipated from the nodes other than cluster heads to send signals to the cluster heads and then the total energy required by the Cluster Head for aggregation and transmit it to base station[1]. Formally speaking, the fitness function used to evaluate individual, in EARP protocol becomes:

$$\varphi_{EAERP}(I^k) = \left(\sum_{i=1}^{nc} \sum_{s \epsilon C_i} E_{TX_{s,CH_i}} + E_{RX} + E_{DA} \right) + \sum_{i=1}^{nc} E_{TX_{CH_i,BS}} \tag{2.11}$$

Here nc is the total number of CHs, s Ci is a non-CHs associated to the ith CH node, ETX node1 and node2 is the energy dissipated for transmitting data from node1 to node2 and ERX and EDA are the energy dissipated for receiving and aggregating data computed as:

$$E_{TX} = E_{elec} * l + E_{fs} * l * d^2, d < d_0 \tag{2.12}$$

$$E_{TX} = E_{elec} * l + E_{mp} * l * d^4, d \geq d_0 \tag{2.13}$$

2.5 Harmony Search Algorithm (HSA)

Harmony Search algorithm are meta heuristic optimization method. This method evolve similar to music improvisation process[7]. In this solution space is our harmony vector which goes on updating up-to the termination point for the better result.

For increasing the longevity and decreasing the energy consumed in WSNs Clustered routing, a new algorithm, called the Harmony Search Algorithm has been developed. The fitness function for the harmony solution is as follows.

$$\varphi_{HSA} = \omega * f_1 + (1 - \omega) * f_2 \tag{2.14}$$

Where

$$f_1 = max_k \left\{ \sum_{\forall nodes \epsilon C_k} d(node_i, CH_k / \|C_k\|) \right\} \tag{2.15}$$

$$f_2 = \sum_{i=1}^{N} E(node_i) / \sum_{j=1}^{k} E(CH_k) \tag{2.16}$$

HSA was created from an inspiration which was based on the musical process where we try to search for a perfect state of harmony which is determined by objective function value[8]. Various steps involved in a HSA-based approach for minimizing the objective function. Two parameters are involved, first is HMCR (Harmony Memory Considering

Rate) and second is Pitch Adjustment Rate (PAR). Harmony memory size can be considered as the solution vectors in the Harmony Memory Matrix. In second step, the initial HM exist of a HMS; number of solutions randomly generated for the purpose of optimization of the problem under consideration. The HMS represented as

$$HM = \begin{pmatrix} I_1^1 & I_2^1 & \cdots & I_k^1 \\ I_1^2 & I_2^2 & \cdots & I_k^2 \\ \vdots & \vdots & \ddots & \vdots \\ I_1^{HMS} & I_2^{HMS} & \cdots & I_k^{HMS} \end{pmatrix} \begin{pmatrix} F^1 \\ F^2 \\ \vdots \\ F^{HMS} \end{pmatrix} \tag{2.17}$$

Step 3 HM is used to improvise a new harmony. HMCR defined in step 1 is used to generate new harmony vector.

$$I'_j \leftarrow \begin{cases} I'_j \epsilon \, HM \; with \; Probability \; HMCR \\ I'_j \epsilon \, I_j \; with \; Probability (1 - HMCR) \end{cases} \tag{2.18}$$

Probability of selecting a part from the HM members can be defined as HMCR and (1-HMCR). If HM will generate, it will also be modified or mutated corresponding to Pitch Adjustment Rate (PAR). The Pitch adjustment is selected by

$$I'_j \leftarrow \begin{cases} I_j^n \epsilon \, HM \; with \; Probability \; PAR \\ I_j^n \; with \; Probability (1 - PAR) \end{cases} \tag{2.19}$$

Step 4 is update new harmony memory. The objective function may have new Harmony vector which is better than objective function value then the new vector can be included in the HM and the worst harmony is excluded. Last step to terminate the criterion.

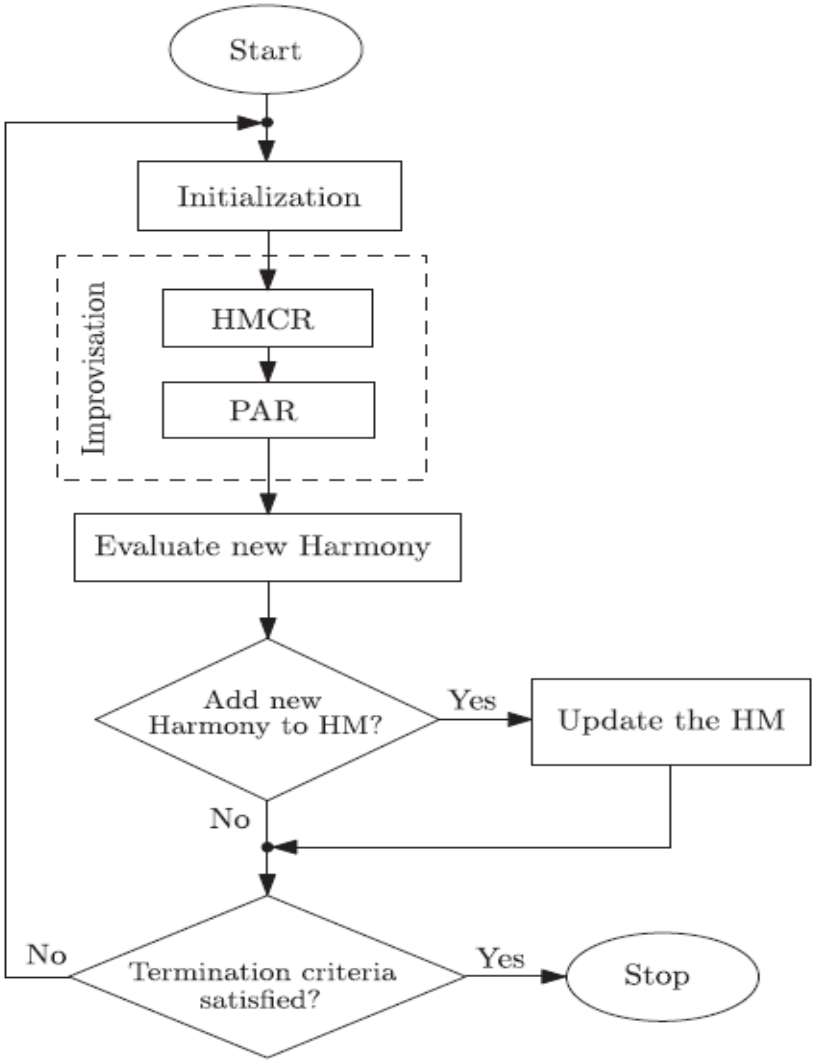

Figure 2.1: Harmony Search Algorithm

2.6 Global-best harmony search algorithm

- First, in Global-best harmony search(GHS), a dynamically updating scheme of parameter PAR which is used in IHS(improvised harmony search). It will improve the performance of GHS and it is explained by the following equation.

$$PAR(FEs) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{maxFEs} FEs \tag{2.20}$$

- where PAR_{FEs}[9] stands for the pitch adjusting rate at generation FEs(function evaluations), and PAR_{min} and PAR_{max} are the rates of adjusting the minimum

and maximum values, , respectively and FEs is the iterative variable and maxFEs represents the number of function evaluations carried out in total.

- The next step is to adjust the pitch step of the HS. This is done by GHS. This is done in order to take advantage of the guiding information of the best harmony in the HM. At the changed step, GHS removes the parameter bw, as it gets difficult to tune it, owing to the fact that it can take on any value between 0 and infinity. GHS introduces a new social term of the best harmony to the HS[9].

- The steps above make it clear that the steps for GHS and Hs are the same, save the exception that the technique involved in providing a new harmony is modified.

2.7 Improved global-best harmony search algorithm

- There are two main parameters, namely, HMCR and PAR that contribute most to the HS along with its variants.

- Thus, in order to get better global and local solutions, a technique can be used for the fine-tuning of the algorithm.

- Also, to ensure that the two parameters dont take on a negative value, a function is used to maintain the sign, which is based on the sine function[9].

$$HMCR(FEs) = HMCR_{min} + \frac{HMCR_{max} - HMCR_{min}}{maxFEs} \times FEs \times max(0, sgn(sin(FEs))) \tag{2.21}$$

- Where $HMCR(FEs)$ is the harmony memory consideration rate for the iterative variable FEs, $HMCR_{min}$ and $HMCR_{max}$ represent the minimum and maximum harmony memory rate respectively. FEs is the variable that can be iterated, and maxFEs denotes the total function evaluations carried out.

- sgn(.) is the function employed to ensure the non-negativity of the two parameters,

and sin(.) is the sine function used for the same.

$$PAR(FEs) = PAR_{min} + \frac{PAR_{max} - PAR_{min}}{maxFEs} \times FEs \times max(0, sgn(sin(FEs))) \tag{2.22}$$

- Where $PAR(FEs)$ is the rate for adjustin the pitch of the iterative variable FEs, PAR_{min} and $PAR_{max}$are understandably, the minimum and maximum adjusting rate, respectively.
- So, it is examined that IGHS is better than GHS and GHS is better than HS Algorithm.

Chapter 3

Multi-objective harmony Search Algorithm

3.1 Multi-objective

Many problems in engineering optimization depends on model based on multi-objective formulation. In number of real-life problems, objectives considered for a problem conflicts with each other. Hence solution optimized with respect to any one objective cannot give desired results because it is not acceptable with respect to the other objectives that conflicts with object considered for optimization. In this kind of multi objective problems, one method is to investigate set of solutions. Each solution is optimized for objectives at acceptable level rather than optimized unfairly for any one objective[10]. We need to consider many objectives while formulating the problems, since the problem cannot be represented completely by a single objective having several number of constraints. If so, there is a vector of objectives,

$$f(x) = [f_1(x), f_2(x), f_3(x), ..., f_n(x)] \tag{3.1}$$

This fomulation must be traded off with respect to each objective. To know importance of objectives relative to each other we need to understand the trade off between multiple objectives and we have to determine systems best capabilities. Understanding trade-off quickly becomes a complex and difficult to quantify problem as the number of objectives increases[11]. Throughout the optimization cycle, designer must rely on

intuition and ability in expressing preferences for each objective. Thus, requirements for a multi-objective design strategy must enable a natural problem formulation to be expressed, and be able to solve the problem and enter preferences into a numerically tractable and realistic design problem.

Multi-objective optimization can be mathematically described as the minimization of a vector of objectives $F(x)$. Each objective has its own constraints and bounds:

$$minf(x),\ Subject\ to\ x \in \tag{3.2}$$

Here $F(x)$ is a vector. As we discussed earlier, components of this vector conflicts with each other. Hence there cannot be unique solution for this kind of problem. Instead, we have to use non-inferiority concept called Pareto Optimality to characterize each of objectives. In non-inferior solution, optimization with respect to one objective results in degrading with respect to another[10]. Now lets define this concept more precisely. In the parameter space, Consider a feasible region, $.x$ is an element in n-dimensional real numbers i.e. $x \epsilon R^n$. X satisfies all constraints and bounds. $x \epsilon R^n$ that satisfies all the constraints i.e.,

$$\Omega = x \epsilon R^n \tag{3.3}$$

This allows definition of the corresponding feasible region for the objective function space:

$$\Lambda = y \epsilon R^n : y = F(X), x \epsilon \Omega. \tag{3.4}$$

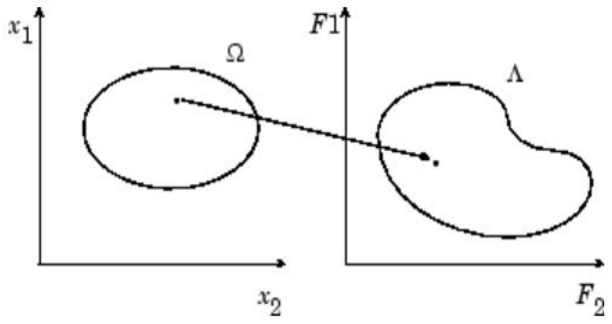

Figure 3.1: Mapping from Parameter Space into Objective Function Space

The performance vector F(x) maps parameter space into objective function space, as represented in two dimensions in the figure Mapping from Parameter Space into Objective Function Space[12].

A noninferior solution point can now be defined.

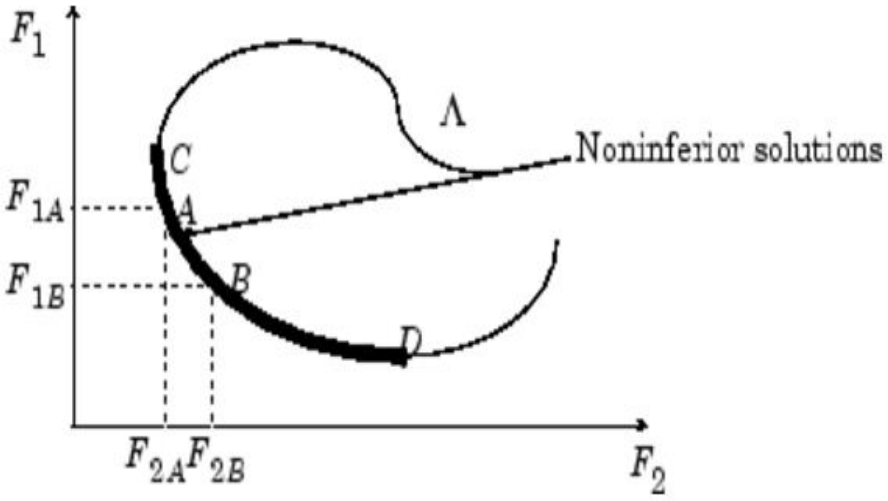

Figure 3.2: Set of Noninferior Solutions

Definition: Point $x* \epsilon \Omega$ can be defined as a non-inferior solution only if for small neighborhood of x* there is not any possible value of ΔXsuch that $(x* + \Delta x) \epsilon \Omega$. In our two-dimensional representation shown in figure set of non inferior solutions lies between C and D. Consider points A and B in figure. These points are non inferior solutions because negative slope at these points means that an improvement in F_1, requires degradation in objective F_2[12].

In figue there are some points where improvement can be attained with respect to all objectives. These points are inferior points. But these points do not carry any value. Therefore, optimization of multi-objective problem is generating and selecting the non-inferior points[11]. Non-inferior solutions are also known as Pareto optima. The goal in multiobjective optimization is construction of the Pareto optima.

3.2 Multi-Objective Harmony Search Algorithm

Most of engineering optimization problems in real world are multi-objective in nature. Reason is these problems have several objectives that needs to be optimized (maximized or minimized depending on preference) at the same time. Most of the times, these objectives are conflicting in nature, that is, improvement in one results in degradation of others. Therefore, for these problems, it is impossible to find one unique optimized solution, which is easy with optimization problems having single objective[7]. Hence, focus in multi-objective problems is finding set of solutions depending on trade-off between multiple objectives. It is left to the expert to choose solution from these set of solution depending on ones preference.

Harmony Search Algorithm (HS) is a recent meta-heuristic. It is inspired improvisation process for music. It was initially aimed as optimizing mono-objective problem. However it has been proven to be effective in many applications in science and engineering. It is proposed to extend the application of HS algorithm for optimizing complex multi-objective problems

Chapter 4

Proposed Work

4.1 Proposals of multi-objective Harmony Search algorithms based on Pareto dominance

- Fundamental difference between the multi-objective variants and the mono-objective HS algorithms proposed in present work is based on solution sorting procedure which can also be known by ranking assignment.

- Associating number to each solution as deemed better or worse than rest, determining an order where the best ones have lower ranking than worse are all part of ranking assignment[8].

- Ranking of solutions in mono-objective problem is determined by objective function. Pareto optimality concept has been incorporated in several methods for multi-objective problems.

- Fonseca and Fleming[8] proposed a ranking assignment method chosen for multi-objective variants of HS algorithm where ranking of solution is determined by x_i for iteration t is defined in equation below:

$$rank(x_i, t) = 1 + p_i^{(t)} \tag{4.1}$$

where $p(t)$ i denotes the number of solutions for the current iteration which dominate the solution in question. So, non-dominated solutions of HM are assigned a ranking

equal to 1, while dominated solutions have a ranking equal to $q \epsilon 2, ..., HMS$.

- To avoid making significant changes in original HS algorithm, multi-objective variant was conceived. Algorithms presents the pseudo code.

Algorithm 1 MOHS

Input: F(x),HMCR,PAR,HMS,MI

Output: P extracted from HM

Randomly initialize HM

while *stopping criterion is not satisfied* **do**

  Improvise a new solution S

  calclulate the Pareto ranking of S considering HM

  if *S has a better ranking than the worst solution in HM* **then**

    Update HM with S

  end

end

Selecting 3 best Solutions

- According to Fonseca Fleming method, difference consists in using HM as repository for best trade-off solutions found at a given point of time and also specify ranking for them.

- Initializing HM with solutions generated randomly is first step of the algorithm and then in the next step it calculates ranking of these solutions by an operation which incurs an asymptotic cost of $O(HMS^2)$ for calculation of ranking and comparing solutions with each other[8].

- Algorithm tries to find a new trade-off solution with each iteration by using decision variable values in HM as considering values which identical to mono-objective HS algorithm.

- For all newly generated solutions, considering solutions stored in HM a ranking is calculated. If these ranking is better than the worst ranking solution stored in HM then new solution replaces the worst one and in case of more than one it chooses any one randomly[13].

- Recalculation of the ranking of solutions in HM incurs an asymptotic cost of $O(HMS)$. The non-dominated solutions stored in HM at each interaction corresponds to approximation of Pareto set for the problem which is to be solved.
- Solutions having ranking equal to one which are stored in HM are returned when stopping criterion is met as best approximation to Pareto optimum for multi-objective problem to be solved.

4.2 Results

- Fobj Values for 50 Nodes(Initially)

F1 * alpha	F2 *(1- alpha)	F3 * 10	Fobj	Rank
4.566667	13.712099	19.353629	37.632395	1
5.633333	13.721127	19.354171	38.708631	2
9.200000	13.617206	19.354225	42.171431	3
11.075000	13.622241	19.355501	44.052742	4
11.706667	13.714644	19.354880	44.776191	5
12.938462	13.767245	19.357555	46.063261	6
13.250000	13.618851	19.354672	46.223523	7
13.230769	13.668968	19.355959	46.255697	8
13.285714	13.671834	19.358720	46.316269	9
15.800000	13.672073	19.359928	48.832000	10

- Fobj Values for 50 Nodes(Middle)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
5.233333	6.213859	9.004209	20.451402	1
8.200000	6.102072	9.004209	23.306281	2
7.933333	7.771901	9.602749	25.307983	3
9.057143	6.687981	9.602749	25.347873	4
9.916667	8.335019	9.885705	28.137391	6
9.916667	8.424484	9.885705	28.226856	7
12.336364	6.620928	9.415699	28.372991	8
12.336364	7.703304	9.415699	29.455367	10
16.476923	6.499200	9.004209	31.980333	11
16.476923	6.499200	9.004209	31.980333	12

- Fobj Values for 50 Nodes(Last)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
10.771875	2.949149	5.833416	19.554441	1
7.033333	4.337205	12.145927	23.516466	2
9.866667	3.976697	10.889806	24.733169	3
10.855172	4.670400	12.145927	27.671500	4
12.996552	4.242517	10.889806	28.128875	6
11.139286	4.719967	12.436511	28.295764	7
13.242857	4.745421	10.889806	28.878084	8
11.846154	5.284832	12.967610	30.098595	9
15.060714	4.956263	12.436511	32.453489	11
14.567742	5.818755	13.935955	34.322451	12

- Fobj Values for 100 Nodes(Initially)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
0.090000	17.035135	17.601527	34.726662	1
1.358000	16.560529	17.437765	35.356294	2
1.330000	16.690312	17.629215	35.649527	3
1.282000	17.095460	17.635556	36.149551	4
1.300000	17.138803	17.630784	36.069587	6
1.937500	16.578072	17.633979	36.149551	7
1.918333	16.943160	17.437765	36.299258	8
1.639000	17.060299	17.632740	36.332039	9
1.639000	17.096318	17.600187	36.335505	10
1.957273	16.828304	17.633061	36.418637	11

- Fobj Values for 100 Nodes(Middle)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
1.922500	7.969515	8.231047	18.123061	1
1.286000	8.892892	8.226441	18.405333	2
1.000000	9.141196	8.655915	18.797111	3
2.610000	8.020641	8.185033	18.815674	4
2.559286	9.098351	8.226441	19.884077	5
2.559286	9.098351	8.226441	19.884077	6
3.790833	8.052207	8.231047	20.074087	7
3.172000	8.836166	8.226441	20.234607	8
3.172000	8.836166	8.226441	20.234607	9
2.232308	10.388161	8.226441	20.846910	10

- Fobj Values for 100 Nodes(Last)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
0.233333	0.940218	6.900692	8.074243	1
0.400000	0.922371	6.900692	8.223063	2
0.400000	0.946744	6.900692	8.247436	3
0.400000	0.987171	6.900692	8.287863	4
0.233333	1.323671	6.900692	8.457696	6
0.650000	1.006254	6.900692	8.556946	7
0.306667	1.394106	6.900692	8.601465	8
0.540000	1.206577	6.900692	8.647270	9
0.540000	1.206684	6.900692	8.647377	10

- Fobj Values for 150 Nodes(Initially)

F1 * alpha	F2*(1-alpha)	F3 * 10	Fobj	Rank
7.328571	14.993821	18.721952	41.044345	1
9.700000	15.106487	18.721898	43.528385	2
15.342857	15.034334	18.720852	49.098043	3
15.990000	15.078682	18.728028	49.796710	4
23.900000	15.179458	18.775930	57.855388	8
26.158333	15.015050	18.722564	59.895947	9
26.525000	15.184884	18.728028	60.437911	10
30.866667	15.057700	18.720852	64.645218	12
32.314286	15.166622	18.722543	66.203450	14
38.811111	15.144276	18.720310	72.675698	18

- Fobj Values for 150 Nodes(Middle)

F1 * alpha	F2*(1-alpha)	F3 * 10	Fobj	Rank
11.250000	7.518066	9.594400	28.362466	1
14.500000	7.584699	9.732392	31.817091	2
16.628571	7.615446	9.732392	33.976410	3
17.700000	7.529572	9.563780	34.793352	4
20.700000	7.584471	9.563780	37.848251	5
22.000000	7.505743	9.563780	39.069522	6
23.288889	7.565949	9.732392	40.587230	8
24.875000	7.639142	9.784353	42.298495	9
26.260000	7.763232	9.732392	43.755624	10
29.250000	7.502905	7.502905	46.485297	12

- Fobj Values for 150 Nodes(Last)

F1 * alpha	F2*(1-alpha)	F3 * 10	Fobj	Rank
12.500000	3.090052	8.427867	24.017919	1
14.633333	3.143768	8.427867	26.204968	2
14.633333	3.143768	8.427867	26.204968	3
20.500000	2.621841	8.427867	31.549708	4
20.500000	3.391518	8.427867	32.319385	5
22.860000	2.663583	8.460511	33.984094	6
22.000000	3.618203	9.798091	35.416294	7
24.400000	3.010876	8.460511	35.871387	8
25.514286	2.934284	8.427867	36.876437	9
26.000000	3.517865	9.798091	39.315956	10

- Fobj Values for 200 Nodes(Initially)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
8.933333	12.424609	19.243144	40.601087	1
17.557143	12.393226	19.238363	49.188732	2
17.914286	12.412850	19.223499	49.550634	3
18.233333	12.428176	19.225628	49.887138	5
19.812500	12.452574	19.225340	51.490414	6
22.500000	12.400195	19.229559	54.129754	7
22.860000	12.424954	19.227328	54.512282	8
24.666667	12.443422	19.244723	56.354812	9
25.075000	12.409678	19.243144	56.727822	10
26.860000	12.448233	19.243144	58.551378	11

- Fobj Values for 200 Nodes(Middle)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
13.175000	6.292510	9.615657	29.083167	1
15.275000	6.139271	9.514113	30.928384	2
15.275000	6.139271	9.514113	30.928384	3
18.071429	6.225852	9.765189	34.062469	4
20.754545	6.195011	9.514113	36.463670	5
22.171429	6.227736	9.735992	38.135157	6
23.012500	6.638438	10.100510	39.751447	7
23.012500	6.638438	10.100510	39.751447	8
24.842857	6.654434	9.750227	41.247518	9
26.410000	6.561536	9.750227	42.721762	10

- Fobj Values for 200 Nodes(Last)

F1 * alpha	F2 *(1-alpha)	F3 * 10	Fobj	Rank
6.500000	4.154599	10.494762	21.149361	1
13.400000	3.138668	6.970812	23.509480	2
11.166667	3.530590	10.076006	24.773262	3
15.200000	3.519106	7.792334	26.511440	4
16.700000	3.274102	6.970812	26.944914	5
18.700000	3.444902	6.843112	28.988014	6
19.850000	3.782969	6.843112	30.476081	8
21.433333	3.740525	6.970812	32.144670	9
22.225000	3.361273	6.970812	32.557085	10
22.225000	3.482459	6.970812	32.678271	11

As pr the above explanation we can see that after giving the rank to the iteration we find the dominate solution.

4.3 Adding Cost Function in HM Fitness Function

$$f_{obj} = \alpha * f_1 + (1 - \alpha) * f_2 + f_3 * 10 \tag{4.2}$$

Where f_3 is

$$f_3 = \left(E_{Avg}(i) * E_{init} \right) / E_{max_{CH}} \tag{4.3}$$

– E(int) which is initial battery level of the node and E(i) is residual energy at sensor node i in the minimum cost function which combines battery level and energy consumption for the cluster head selection. Energy consumed at node i is determined by E(Avg).

– Cluster head with highest energy will be selected by minimum cost function algorithm in order to increase network lifetime and due to these cluster heads which has and high residual energy gets selected.

Chapter 5

Implementation Results

This chapter covers details of energy usage by the nodes, cluster-heads and by cluster during the rounds.

5.1 Radio Energy Dissipation Model

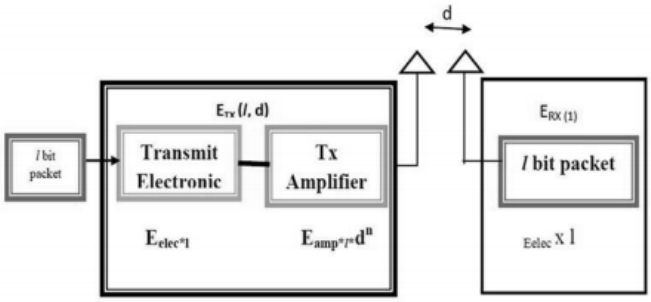

Figure 5.1: Radio model

At transmitter side, energy E_{T_x} is required to transmit L bits at a distance d.

$$E_{TX}(L,d) = \begin{cases} L(E_{elec} + \varepsilon_{fs} d^2) & d < d_0 \\ L(E_{elec} + \varepsilon_{mp} d^4) & d \geq d_0 \end{cases} \tag{5.1}$$

Here d_0 is the deciding factor. It can be use as free space propagation model or multi-path radio propagation model. ε_{fs} and ε_{mp} are the amplification component. It's depend on the propagation model. At receiver side, energy is required to receive

L bits,

$$E_{RX}(L) = LE_{elec} \tag{5.2}$$

Energy required for one round,

$$E_{round} = L(2NE_{elec} + NE_{DA} + K\varepsilon_{mp}d^4_{toBS} + N\varepsilon_{fs}d^2_{toCH}) \tag{5.3}$$

Where E_{DA} is energy required to aggregate data, d_{toBS} and d_{toCH} is distance to base station and distance to cluster head respectively and N is number of nodes.

5.2 Simulation Parameters

We performed simulation of protocols using ns2 simulator with C++ and tcl. The simulations were performed by varying field size, number of nodes as mentioned in table 5.1. Base station location was also varied to check the impact of this factor on simulations. The network topologies were created with random node deployments, which are mostly seen in literature.

Parameter	**Value**
Node distribution	Random
BS location	Center
No. of Nodes	50,100,150,200
Initial Node Energy	2J
Simulation Time	3600s
Bandwidth of the channel	1 Mbps
Packet header size	25 Bytes
Message size	500 Bytes

Table 5.1: Simulation Parameter

The figure 5.2 displays the graph of alive nodes vs time(Sec). The simulation is taken in consideration for 150 nodes. The graph represents two major algorithms which are HSA and MHSA. 1st lowest is considered in MHSA and the readings of the 2nd lowest and 3rd lowest are plotted in the graph. The results display that the MHSA shows better results than the HSA as the nodes in MHSA are staying alive for a longer time than the HSA.

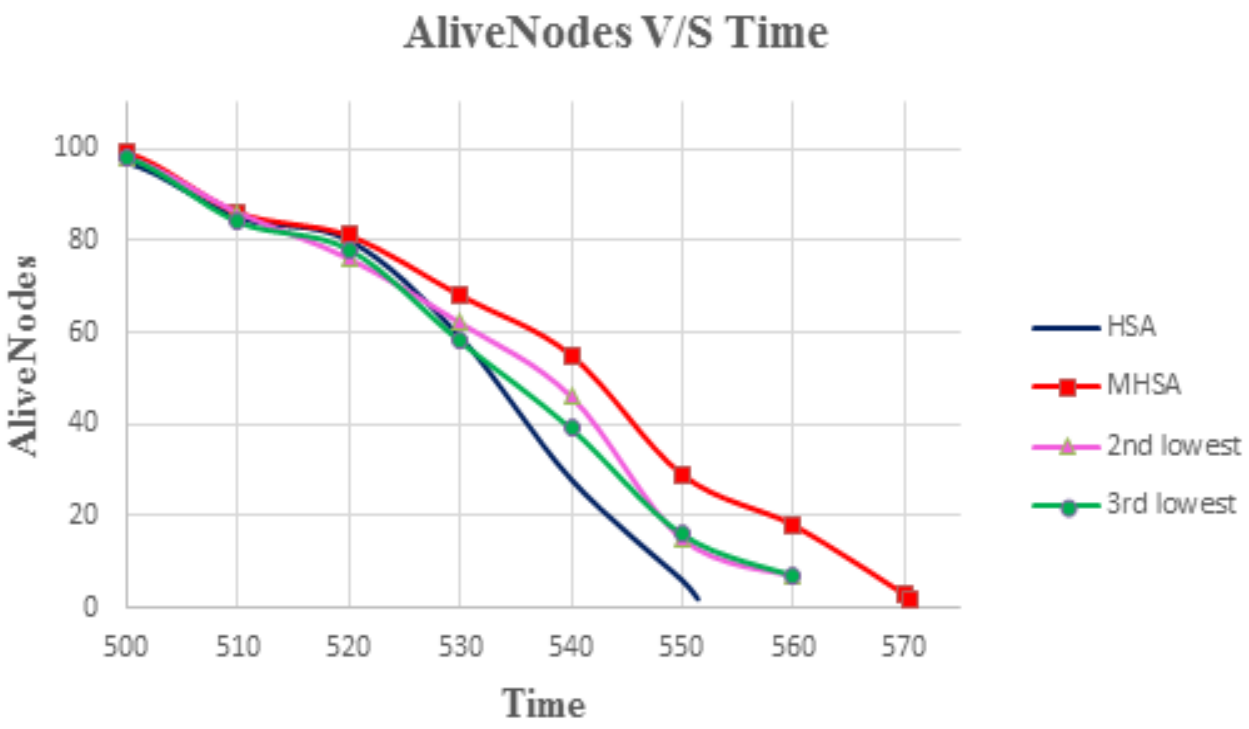

Figure 5.2: For 150 Nodes

The figure 5.3 displays the graph of data(packets) vs time(Sec). The simulation is taken in consideration for 150 nodes. The graph represents two major algorithms which are HSA and MHSA. 1st lowest is considered in MHSA and the readings of the 2nd lowest and 3rd lowest are plotted in the graph. The results display that the 3rd lowest results give the highest data transmission whereas the MHSA and its 2nd lowest gives better results than the standard HSA.

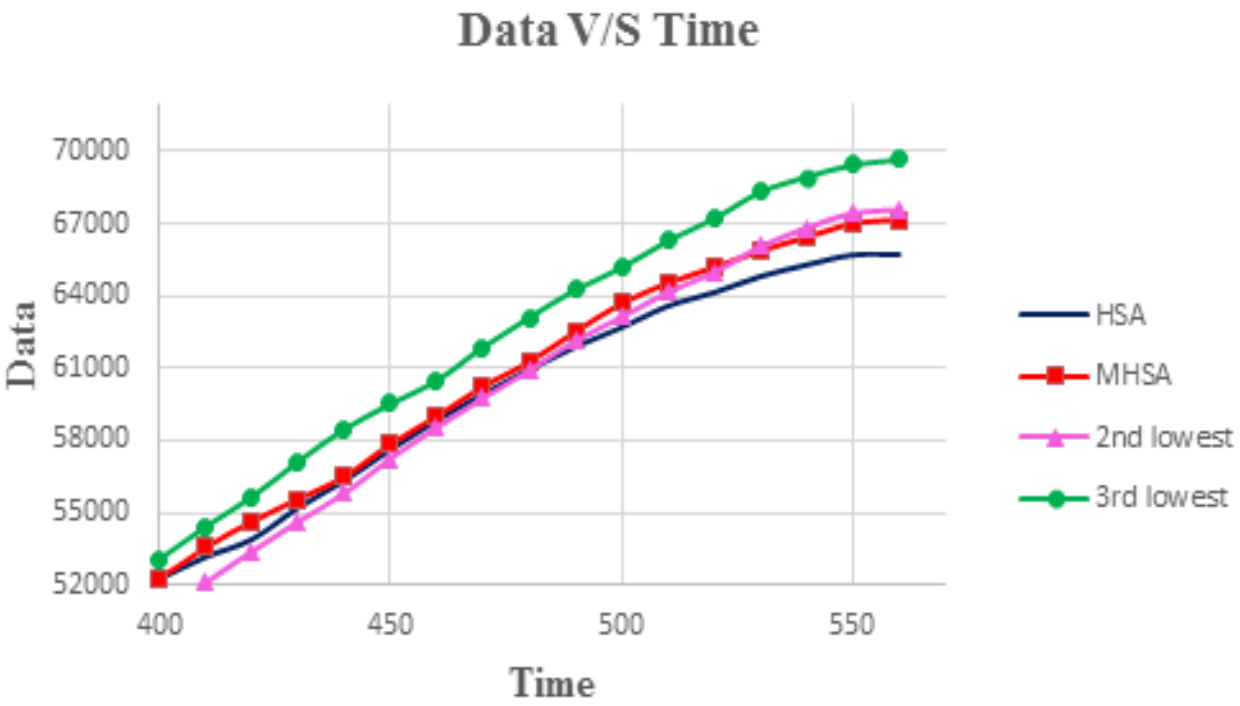

Figure 5.3: For 150 Nodes

The figure5.4 displays the graph of energy(joule) vs time(Sec). The simulation is taken in consideration for 150 nodes. The graph represents two major algorithms which are HSA and MHSA. 1st lowest is considered in MHSA and the readings of the 2nd lowest and 3rd lowest are plotted in the graph. The results clearly show that the energy dissipation in MHSA is better compared to the other algorithms.

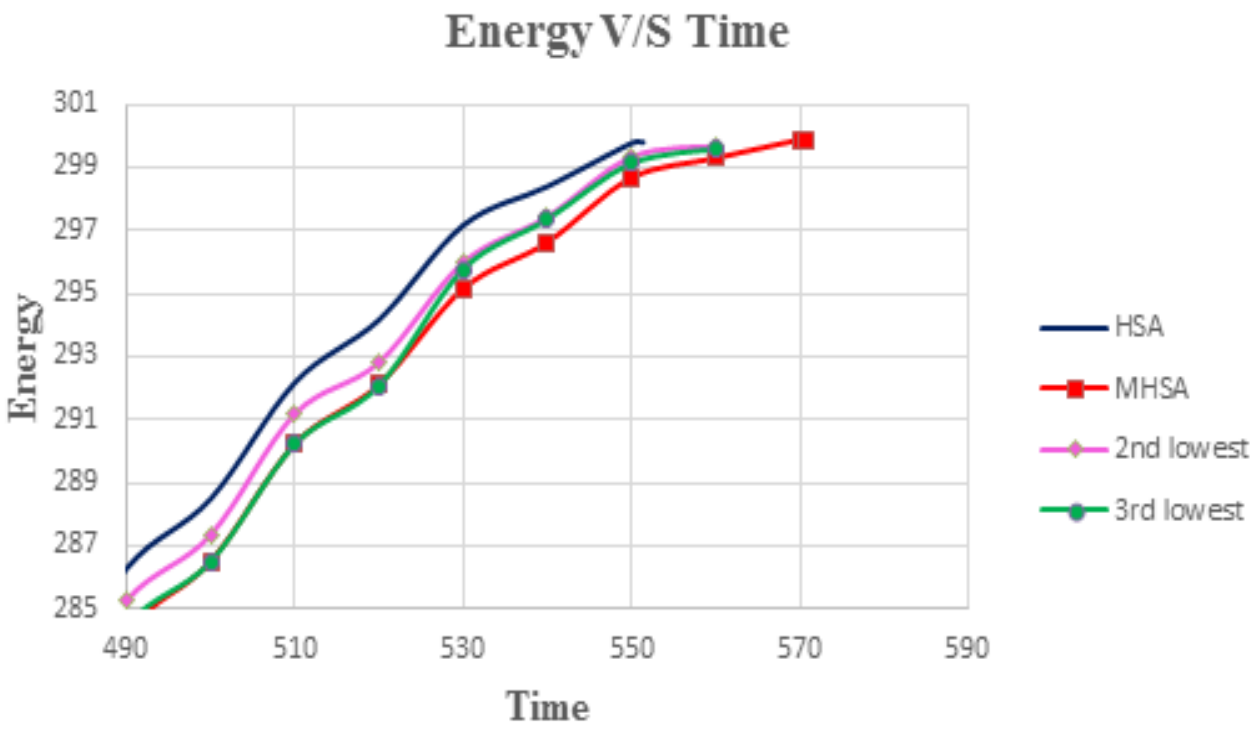

Figure 5.4: For 150 Nodes

The figure 5.5 displays the graph of alive nodes vs time(Sec). The simulation is taken in consideration for 200 nodes. The graph represents two major algorithms which are HSA and MHSA. 1st lowest is considered in MHSA and the readings of the 2nd lowest and 3rd lowest are plotted in the graph. The results display that the MHSA shows better results than the HSA as the nodes in MHSA are staying alive for a longer time than the HSA.

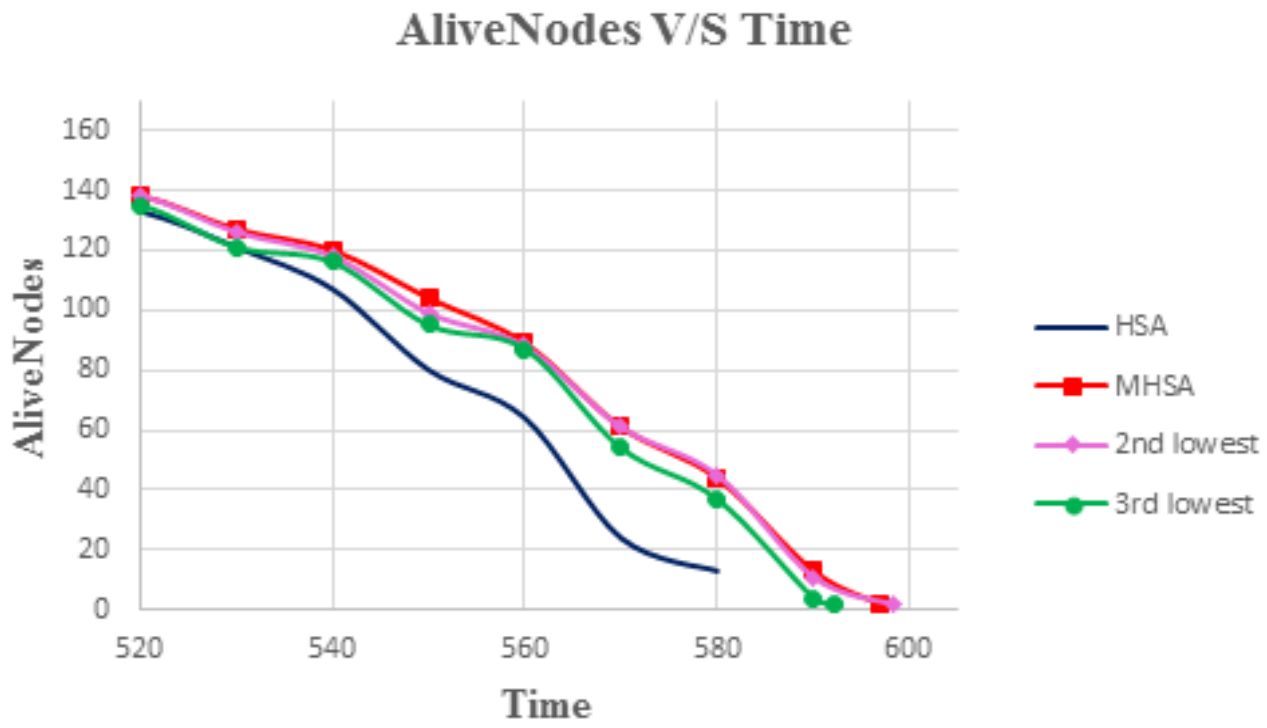

Figure 5.5: For 200 Nodes

The figure 5.6 displays the graph of data(packets) vs time(Sec). The simulation is taken in consideration for 200 nodes. The graph represents two major algorithms which are HSA and MHSA. 1st lowest is considered in MHSA and the readings of the 2nd lowest and 3rd lowest are plotted in the graph. The results display that the 3rd lowest results give the highest data transmission whereas the MHSA and its 2nd lowest gives better results than the standard HSA.

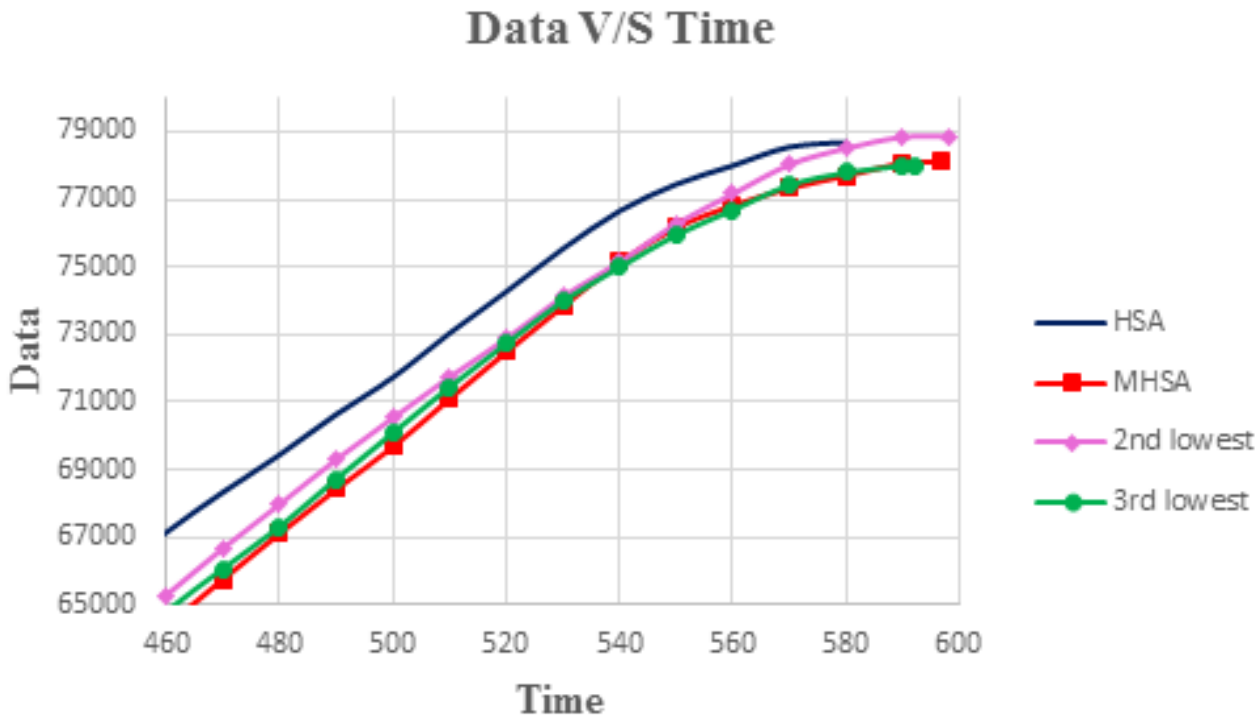

Figure 5.6: For 200 Nodes

The figure5.7 displays the graph of energy(joule) vs time(Sec). The simulation is taken in consideration for 150 nodes. The graph represents two major algorithms which are HSA and MHSA. 1st lowest is considered in MHSA and the readings of the 2nd lowest and 3rd lowest are plotted in the graph. The results clearly show that the energy dissipation in MHSA is better compared to the other algorithms.

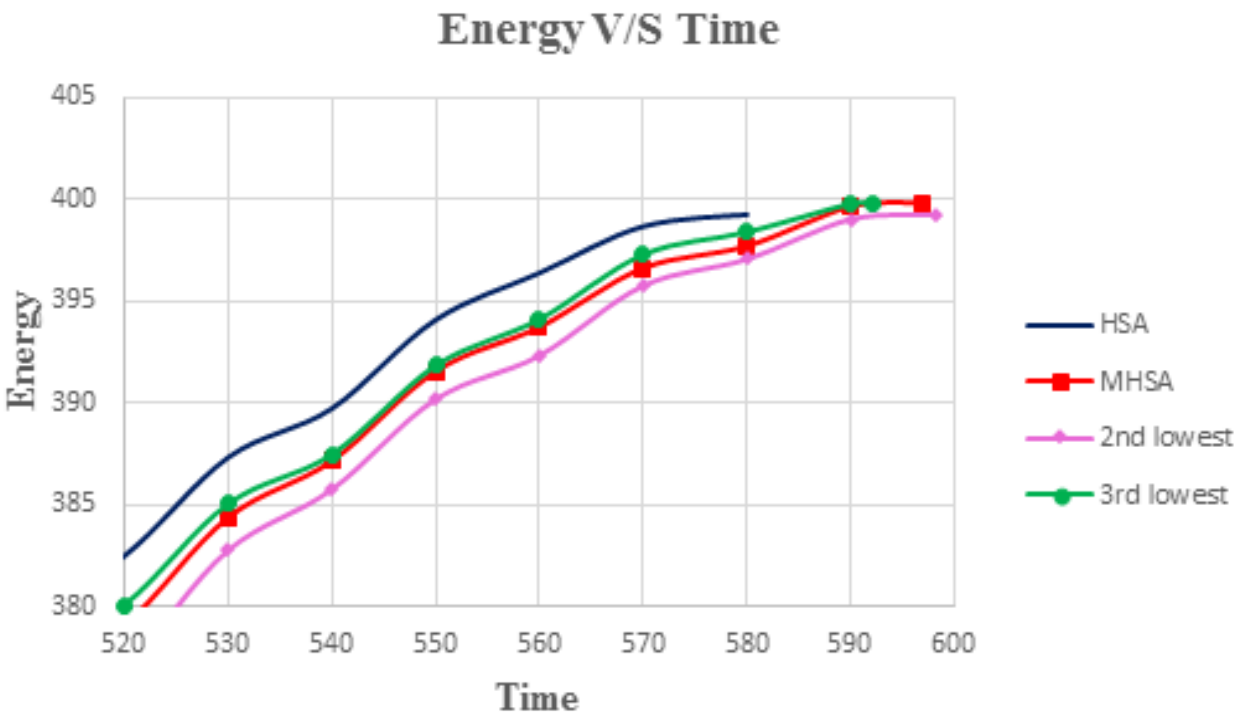

Figure 5.7: For 200 Nodes

Protocol	CPU Time(in ms)			
	50nodes	**100nodes**	**150nodes**	**200nodes**
HSA	46.0	319.0	1136.0	233.0
MHSA	5.2	23.0	72.0	133.0

Table 5.2: CPU Timing

Protocol	Real Time(in ms)			
	50nodes	**100nodes**	**150nodes**	**200nodes**
HSA	45.4	319.6	114.0	233.5
MHSA	4.0	24.2	69.8	134.6

Table 5.3: Real Time

While clusters are selecting their CH nodes, we are calculating Cpu time and Real time for HSA and MHSA. It show that HSA is taking maximum time than MHSA algorithm.

Chapter 6

Conclusion

Mostly optimization problems are multi-objective being of mixed nature either minimization or maximization. Usually, these objective are in conflict. For this reason, it is not always possible to find a unique optimal solution for these problems. Here we are finding best 3 solution among the various solutions. So finding best solution, we are comparing existing HSA with MHSA. MHSA performs efficiently as compared to HSA as the multiple objectives are considered and multiple solutions are declared as possible solution.

Bibliography

[1] A. A. Baraa and E. A. Khalil, "A new evolutionary based routing protocol for clustered heterogeneous wireless sensor networks," *Applied Soft Computing*, vol. 12, no. 7, pp. 1950–1957, 2012.

[2] D. Hoang, P. Yadav, R. Kumar, and S. Panda, "A robust harmony search algorithm based clustering protocol for wireless sensor networks," in *Communications Workshops (ICC), 2010 IEEE International Conference on*, pp. 1–5, IEEE, 2010.

[3] E. A. Khalil and A. A. Baraa, "Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks," *Swarm and Evolutionary Computation*, vol. 1, no. 4, pp. 195–203, 2011.

[4] D. C. Hoang, P. Yadav, R. Kumar, and S. K. Panda, "Real-time implementation of a harmony search algorithm-based clustering protocol for energy-efficient wireless sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 10, no. 1, pp. 774–783, 2014.

[5] E. A. Khalil, *An Evolutionary Routing Protocol for Dynamic Clustering of Wireless Sensor Networks*. PhD thesis, College of Science, Baghdad University, 2008.

[6] H. Gou, Y. Yoo, and H. Zeng, "A partition-based leach algorithm for wireless sensor networks," in *Computer and Information Technology, 2009. CIT'09. Ninth IEEE International Conference on*, vol. 2, pp. 40–45, IEEE, 2009.

[7] L. M. Pavelski, C. P. Almeida, R. Gonçalves, *et al.*, "Harmony search for multi-objective optimization," in *Neural Networks (SBRN), 2012 Brazilian Symposium on*, pp. 220–225, IEEE, 2012.

[8] J. Ricart, G. Hüttemann, J. Lima, and B. Barán, "Multiobjective harmony search algorithm proposals," *Electronic Notes in Theoretical Computer Science*, vol. 281, pp. 51–67, 2011.

[9] W.-l. Xiang, M.-q. An, Y.-z. Li, R.-c. He, and J.-f. Zhang, "An improved global-best harmony search algorithm for faster optimization," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5788–5803, 2014.

[10] Mathworks, "Multiobjective optimization : http://in.mathworks.com/help/gads/what-is-multiobjective-optimization.html."

[11] K. Deb, "Multi-objective optimization," in *Search methodologies*, pp. 403–449, Springer, 2014.

[12] O. Grodzevich and O. Romanko, "Normalization and other topics in multi-objective optimization," 2006.

[13] Z. W. Geem, "Multiobjective optimization of water distribution networks using fuzzy theory and harmony search," *Water*, vol. 7, no. 7, pp. 3613–3625, 2015.

[14] C.-W. Bong and M. Rajeswari, "Multi-objective nature-inspired clustering and classification techniques for image segmentation," *Applied Soft Computing*, vol. 11, no. 4, pp. 3271–3282, 2011.

[15] Z. W. Geem, "Multiobjective optimization of time-cost trade-off using harmony search," *Journal of Construction Engineering and Management*, 2009.

[16] Y. Ding, S. Gregov, O. Grodzevich, I. Halevy, Z. Kavazovic, O. Romanko, T. Seeman, R. Shioda, and F. Youbissi, "Discussions on normalization and other topics in multiobjective optimization," in *Fields-MITACS, Fields Industrial Problem Solving Workshop*, 2006.

[17] Z.-R. Peng, H. Yin, H.-T. Dong, H. Li, and A. Pan, "A harmony search based low-delay and low-energy wireless sensor network," *Network*, vol. 1, p. 1, 2015.

[18] M. Karimi, H. R. Naji, and S. Golestani, "Optimizing cluster-head selection in wireless sensor networks using genetic algorithm and harmony search algorithm," in *Electrical Engineering (ICEE), 2012 20th Iranian Conference on*, pp. 706–710, IEEE, 2012.

[19] A. A. Abbasi and M. Younis, "A survey on clustering algorithms for wireless sensor networks," *Computer communications*, vol. 30, no. 14, pp. 2826–2841, 2007.

[20] W. Naruephiphat and C. Charnsripinyo, "Clustering techniques in wireless sensor networks," in *New Trends in Information and Service Science, 2009. NISS'09. International Conference on*, pp. 1273–1278, IEEE, 2009.

[21] G. Raval, M. Bhavsar, and N. Patel, "Analyzing the performance of centralized clustering techniques for realistic wireless sensor network topologies," *Procedia Computer Science*, vol. 57, pp. 1026–1035, 2015.

[22] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: recent developments and deployment challenges," *Network, IEEE*, vol. 20, no. 3, pp. 20–25, 2006.