Energy Efficient Genetic Algorithm Model for Wireless Sensor Networks

N. Thangadurai, Dr. R. Dhanasekaran, and R. Pradeep

***Abstract*—** Wireless communication has enable to develop minimum energy consumption clustering protocols and various algorithms for wireless sensor networks (WSN). One of the important issues in WSN belongs to the limited battery of sensor nodes. When sensor nodes are placed in a field, it is difficult to replace their batteries. To minimize the energy consumption of nodes WSN introduce various protocols and algorithms to minimize the energy consumption. Genetic algorithms (GAs) are powerful search techniques that are used successfully to solve problems in different aspects. GAs is easy to implement and provide substantial gains in performance. This paper mainly focuses on various models and different types of crossover implementations of existing GAs and we proposed a new GA model to minimize total energy consumption and provide better energy efficiency compare to other existing GAs techniques to extend the lifetime of networks.

***Keywords*—**Genetic Algorithm, Hierarchical Algorithm, Parallelization methods

I. Introduction

GENETIC Algorithms (GAs) are efficient based on principles of natural selection and genetics. They applied successfully to find acceptable solutions to multiple problems in business, engineering, and science [1]. GAs will find better solutions in reasonable amount of time, but as they applied to bigger problems, but there is an increase in the time required to find solutions. Our aim is to collect and present important methods based on GAs. To organize the growing amount of literature in this area, the paper presents a classification of the different types of implementations of GAs. The most popular GAs consists of multiple populations that evolve separately every time and exchange individuals occasionally. This kind of GAs is called multi-deme, coarse-grained or distributed GAs, and this survey concentrates on this class of algorithm. However, this paper describes the major types of GAs and discusses briefly about few examples. This paper explains in detail how GA works, but it also mention few applications to show that GAs are useful in the real world scenarios.

N. Thangadurai, Research Scholar, Research & Development Centre, Bharathiar University, India. (mrgolddurai@gmail.com)

Dr. R. Dhanasekaran, Director-Research, Syed Ammal Engineering College, Ramanathapuram, India. (rdhanashekar@yahoo.com)

R. Pradeep, M. Tech -Telecommunication Networks, SRM University, Chennai, India. (pradeep.r149@gmail.com)

II. Genetic Algorithm

This section provides basic idea on Gas, and defines some terms and describes how a simple GA works. Genetic algorithms are search algorithms based on natural selection and recombination process [2]. They attempt to find a solution to the problem at hand by manipulating a population of candidates. The population is evaluated and the best solutions are selected to reproduce to form the next generation. In number of generations, good traits dominate the total population, resulting in better quality of the solutions. The basic ideas in GAs are Darwinian evolution: bad traits will be eliminated from the population which do not survive the process of selection. The good traits survive and mixed by mating to form better individuals. Mutation also there in GAs, but considered a secondary operator. This ensures diversity is not lost in the population, so GA can continue to explore the possibilities [3]. They formalized the concept of building blocks which are string templates, matches a portion of the individuals and act as a unit to influence the fitness function of individuals. The prevailing theory proposes that GAs work by propagating selection and crossover.

We follow Goldberg, Deb, and restrict the notion of a building block to the shortest schemata that contribute to the global optimum [1]. In this view, a position of two BBs of order at a particular string does not lead to a BB of order, but instead to two separate BBs. Often, the individuals are composed of a binary string of a fixed length, and thus GAs explore a search space formed by points. The population consists of points chosen randomly, unless there is a heuristic to generate good solutions for the particular domain. Later a portion of the population is still generated randomly to ensure the diversity in the solution. The size of the population is important since it influences whether the GA is able to find good solutions and the time it takes to reach them. If the population is small, there might not be an adequate supply of building blocks, and will be difficult to identify proper solutions. If the population is big, GA will waste computational resources by processing unnecessary individuals. This balances the quality and the time GA needs to find a solution. Each individual in the population has its own fitness function.

The fitness is a measure that how well the individual solves a problem. GAs often used as optimizers, and the fitness of an individual is the value, that objective function at the point represented by a binary string. Selection uses the fitness value to identify the individuals that will reproduce and mate to produce the next generation. Simple GAs use two different operators based on natural genetics to explore search space: crossover and mutation. Crossover is the primary exploration mechanism in Gas, which takes two random individuals from the already selected to form the next generation and exchanges random sub-strings between them.

III. EXISTING GA'S

The idea behind GA's to divide a task into small sub-tasks and solve the small problems simultaneously using multiple processors [4]. This divide-and-conquer approach applied to GAs in different ways. Few parallelization methods use a single population and others divide the population into multiple isolated sub-populations. Few methods can exploit massively parallel computer architectures and others are better suited to multi-computers with fewer and more powerful processing elements. There are three main types of GAs: (1) Global single-population master slave GAs, (2) Single-population fine-grained and (3) Multiple-population coarse-grained GAs. A master-slave GA uses a single parametric population, but the evaluation of fitness is distributed among multiple processors (see Fig.1). In this GA, selection and crossover considers the entire population, it is also known as global parallel GAs. Fine-grained GAs suited for massively parallel computers and consist one spatially-structured population. Selection and mating are restricted to neighbourhoods, but overlap permits some interaction among all the individuals (see Fig.2). The ideal case is to have one individual for one processing element.

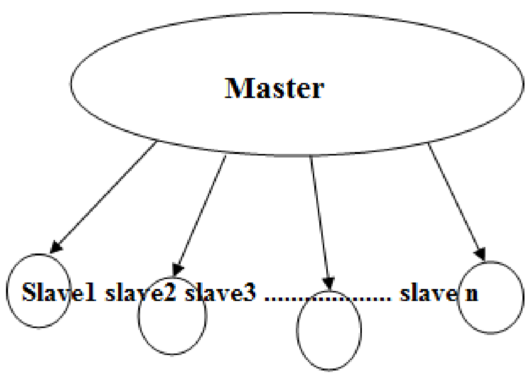

Fig. 1 A simple Master-Slave GA

The slaves evaluate the fitness of the individuals. Multiple-population Gas are more sophisticated, as they consist on several sub-populations which exchange individuals occasionally (Fig.3). This exchange of individuals called migration and it is controlled by several parameters. Multiple-deme GAs is very popular, but also the class of GAs which is most difficult to understand. Multiple-deme GAs introduces fundamental changes in the operation of the GA and has a different behaviour than simple GAs. Multiple-deme GAs are known with different names. Sometimes they are known as "distributed" GAs, because they are usually implemented on distributed memory MIMD computers. Since the computation to communication ratio is high, they are occasionally called coarse-grained GAs. Multiple-deme GAs resembles the island model in Genetics which considers isolated demes and the GAs are also known as island GAs. The size of the demes is smaller than the population used in a serial GA, and the parallel GA converges faster than serial GA. When we compare the performance of the serial and the parallel algorithms, have to consider the quality of the solutions found in different case. While it is true that smaller demes converge faster, and the quality of the solution might be poorer.

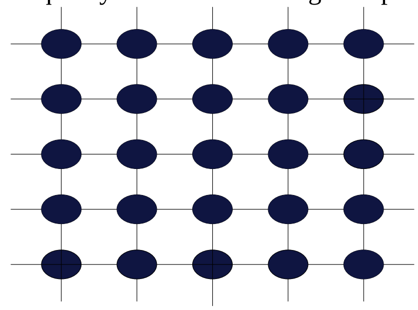

Fig. 2 A schematic diagram of a fine-grained GA.

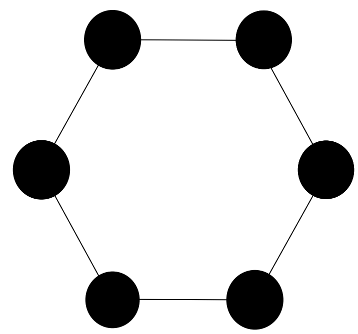

Fig.3 A schematic of a multiple-population GA

It is important to emphasize that the master-slave parallelization method does not affect the behaviour of the algorithm, and last two methods change the way of GA works. In master-slave GAs, selection considers all the population, but in the other two GAs, selection considers a subset of individuals. The master-slave GAs goes with random mating of any two individuals in the population, but the other methods mating has restricted to a individuals of subset. To parallelize Gas, the method combines multiple demes with master-Slave GAs or fine-grained GAs. This hierarchical GA algorithm has some spatially distributed population, and implemented very efficiently on massively parallel computers. Each process is a simple GA, and there is (infrequent) communication between the populations higher level they are multiple-deme algorithms with single-population GAs (either master-slave or fine-grained) at the lower level. A hierarchical GAs combines the benefits of its components, and it promises better performance than any of them alone.

A. Tree Genetic algorithm

In this tree type GA the nodes uses the shortest path algorithm to find the source to the destination so the fitness of the genetic algorithm is maintained this tree type genetic algorithm is also known as two-tiered wireless sensor network [5-7].

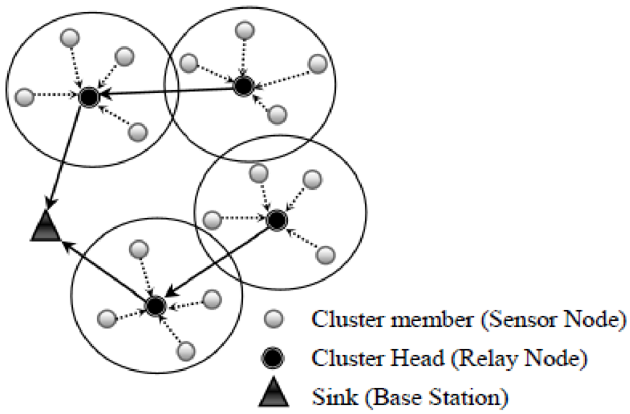

Fig.4 An example of a two-tiered WSN

The above is an two tiered cluster is being used for the routing in this the minimum spanning tree is used for forming the two different routes to find the sink in this it uses the shortest path algorithm to reach the destination [8].

B. Global GA

This section explains the global parallelization method. This algorithm uses a single population and the evaluation of the individuals and the applications of genetic operators are done in parallel process. In the serial GA, each individual may compete and mate with any other individual. Global GAs is usually implemented as master-slave programs; with the master stores the population and the slaves evaluate the fitness. The most common operation is the evaluation of the individuals, because the fitness of an individual is independent from the rest of the population and no need to communicate in this phase.

Evaluation of individuals is parallelized by assigning a fraction of the total population to each of the processors available in the network. Communication occurs only when each slave receives its subset of individuals to evaluate and the slaves return the fitness values [9][10][11]. When the algorithm stops and waits to receive the fitness values of all the population before proceeding to the next generation, then the algorithm is synchronous.

A synchronous master-slave GA has the same properties as a simple GA; with the operation speed will be the only difference. It is also possible to implement an asynchronous master-slave GA where the algorithm does not stop or wait for any slow processors, but it does not work like a simple GA.

This section explains the global parallelization method. This algorithm uses a single population and the evaluation of the individuals and the applications of genetic operators are done in parallel process. In the serial GA, each individual may compete and mate with any other individual. Global GAs is usually implemented as master-slave programs; with the master stores the population and the slaves evaluate the fitness. The most common operation is the evaluation of the individuals, because the fitness of an individual is independent from the rest of the population and no need to communicate in this phase.

Evaluation of individuals is parallelized by assigning a fraction of the total population to each of the processors available in the network. Communication occurs only when each slave receives its subset of individuals to evaluate and the slaves return the fitness values [9][10][11]. When the algorithm stops and waits to receive the fitness values of all the population before proceeding to the next generation, then the algorithm is synchronous.

A synchronous master-slave GA has the same properties as a simple GA; with the operation speed will be the only difference. It is also possible to implement an asynchronous master-slave GA where the algorithm does not stop or wait for any slow processors, but it does not work like a simple GA

C. Hierarchical Genetic Algorithms

A few researchers have tried to combine two of the methods to parallelize Gas, producing hierarchical GAs. Some of these new hybrid algorithms add a new degree of complexity to the already complicated scene of GAs, but other hybrids manage to keep the same complexity as one of their components. When two methods of GAs are combined they form a hierarchy [12]. At the upper level most of the hybrid GAs are multiple-population algorithms. Some hybrids have a fine-grained GA at the lower level (see Fig.5).

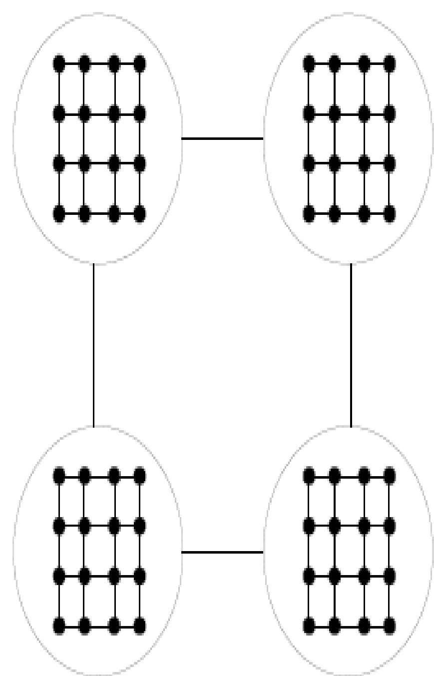

Fig.5 Hierarchical GA combines a multi-deme GA and a fine-grained GA

At the lower level the migration rate is faster and the communications topology is much denser than at the upper level [13]. This method of hybridizing GAs is to use multi-deme GAs at both the upper and the lower levels. The idea is to force panmictic mixing at the lower level by using a high migration rate and a dense topology, while a low migration rate is used at the high level. The complexity of this hybrid would be equivalent to a multiple-population GA if we consider the groups of panmictic sub-populations as a single deme.

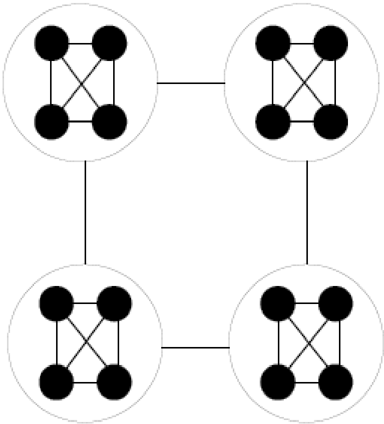

Fig. 6 Hybrid uses multi-deme GAs

IV. PROPOSED GENETIC ALGORITHM MODEL

This model consist of two phases namely set-up and state phase for each round in the protocol. In preparation state all nodes initially performs cluster head selection process and send all the packets with status of a cluster head (or) node position to one base station before the set-up phase of the first round. After receiving all the data from the nodes the base station start searching for the cluster heads through GA by minimizing the total energy consumption required for the competition of one round. Cluster Head selection process can be done by set-up phase and the transmitting data from Cluster Head to Base Station can be done by state phase for all rounds and minimize the total energy consumption of all nodes and extend the network life time.

A. Communication between the nodes

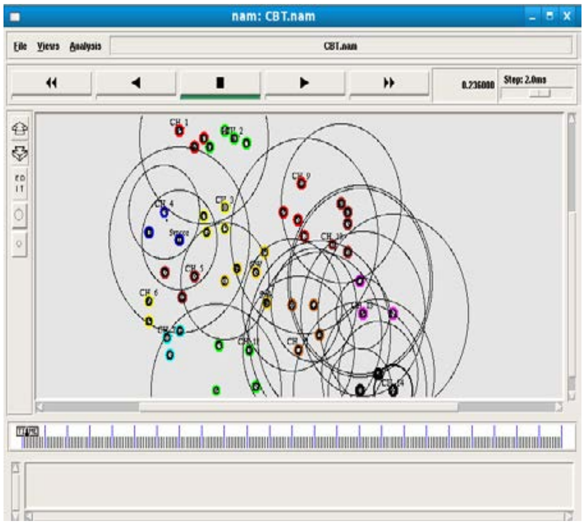

Fig.7 Communication between the nodes

In this phase the each and every nodes in the network is being communicated to each other to indicate all the nodes are alive and they are ready for aggregating the data from the surrounding.

B. Communication between the clusters

We can observe from the above simulation that the CH4 senses the data and then it sends the sensed data to its neighbouring cluster head CH3 and then the data is being sent to gateway through CH3.

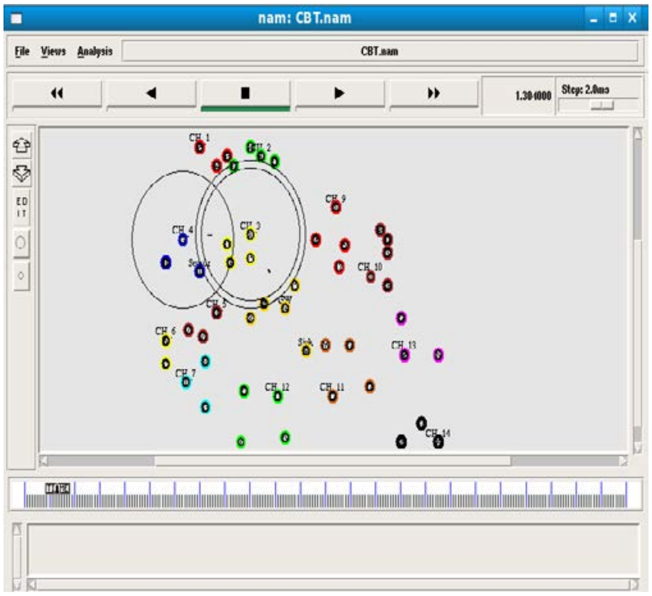

Fig. 8 Communication between the clusters

C. Movement of nodes

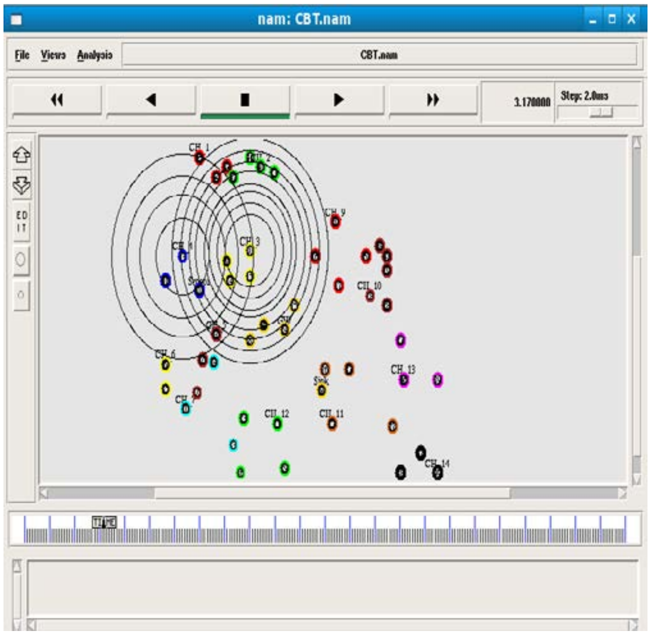

Fig.9 Movement of nodes

In this we can observe the movement of nodes from one cluster to the other, from CH7 the nodes are moved to CH12 to form a cluster and the cluster head moves towards CH6 to form a cluster with CH6.

D. Data aggregation at the base-station

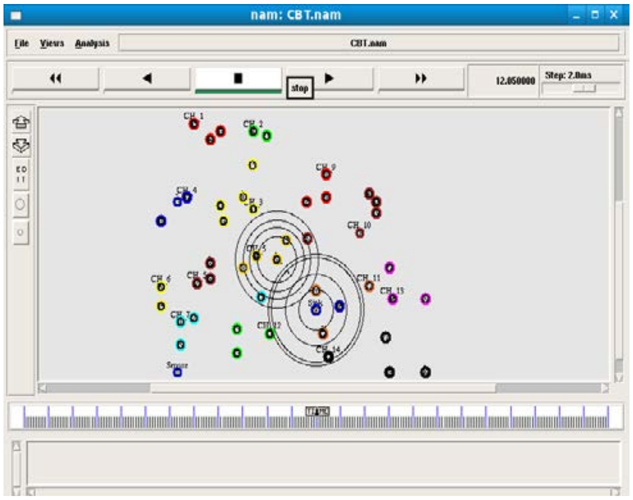

Fig.10 Aggregation of data

All the data which collected by the nodes are being delivered to the gateway and the gateway sends the aggregated data to the sink i.e. the base station in between this process the nodes are moved from one cluster to another cluster.

IV. CONCLUSION

In this paper we discussed about Genetic Algorithm and some types of genetic algorithm and their fitness in the network. We have seen how the crossover is used to increase the stability of the networks. We are working on comparing the results with the existing GA methods to show that the new GA model will give better energy consumption and extends the life time of Wireless Sensor Networks.

REFERENCES

[1] D. E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989.

[2] M. Mitchell, An Introduction to Genetic Algorithms. MIT Press, Cambridge, MA, 1996.

[3] S. Jin, M. Zhou, A.S. Wu, Sensor network optimizationUsing a genetic algorithm, in: 7th World Multi conference on Systemics, Cybernetics and Informatics, Orlando, FL, 2003.

[4] 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), pp. 97–125, Kluwer, 2005.

[5] Gholam Hossein EkbataniFard, Reza Monsefi "A Fast Multi-objective Genetic Algorithm based Approach for Energy Efficient QoS-Routing in Two-tiered Wireless Multimedia Sensor Networks" Modern Applied Science Vol. 4, No. 6; June 2010

[6] Ataul Bari, Shamsul Wazed, Arunita Jaekel, Subir Bandyopadhyay "A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks", Ad Hoc Networks Vol.7, Issue4, June 2009, Pages 665-676.

[7] A. Bari, A. Jaekel, S. Bandyopadhyay, "Maximizing the lifetime of two tiered sensor networks", in: the Proceeding of IEEE International Electro/Information Technology Conference (EIT 2006), MI, May 2006, pp. 222–226.

[8] Dr.Sajid Hussain, Obidul Islam "Genetic Algorithm for Energy Efficient Trees in Wireless Sensor Networks".

[9] Afsane Zahmatkesh, Mohammad H. Yaghmaee "A Genetic Algorithm-Based Approach for Energy- Efficient Clustering of Wireless Sensor Networks" 2011 International Conference on Network Communication and Computer (ICNCC 2011)

[10] A. Zahmatkesh and M. H. Yaghmaee" A Genetic Algorithm-Based Approach for Energy-Efficient Clustering of Wireless Sensor Networks" *International Journal of Information and Electronics Engineering, Vol. 2, No. 2, March 2012.*

[11] Sudakshina Dasgupta, Paramartha Dutta" An Energy Efficient Genetic Approach for Clustering of Wireless Sensor Network", International Journal of Information Engineering IJIE Vol.2, Iss2, PP 54-58, Jun2012.

[12] S.Hussain, A.Matin and O.Islam, "Genetic Algorithm for Hierarchical wireless Sensor Network", Journal of Networks, Vol. 2, No. 5, 2007, pp. 87-97.

[13] Manjeshwar A. and D. P. Agarwal, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks", Proc. of the IEEE IPDPS, Fort Lauderdale, USA, Apr. 2002, pp. 195-202