ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

Design of Memetic Algorithm to Enhance Coverage in Wireless Sensor Networks with Minimum Number of Sensors

Channakrishnaraju[1], Dr. M.Siddappa[2]

Research Scholar, Sri Siddhartha Academy of Higher Education [SSAHE], Tumkur, India[1]

Professor & Head, Department of CS & SSIT, Tumkur, India[2]

Abstract: Wireless sensor networks are a rapidly growing area for research & development. The sensor coverage problem is prime factor being considered in recent advances. The coverage concept is subject to a wide range of interpretations due to different types of sensors and their applications. Genetic algorithm determines the optimal combination of sensor to achieve an objective of an efficient coverage and connectivity of sensors. The Genetic Algorithms restricts Reproduction, Crossover, and Mutation Operation to Sensors. Memetic Algorithm is Type Of Genetic Algorithms used to enhance the coverage and minimizes the number of sensors should be deployed , with minimum number of sensors efficient coverage is possible by eliminating the inefficient sensors by using population best fit method is used in this paper. To obtain the Memetic algorithm we also discuss about set k cover problem to identify which sensors are having efficient coverage and NP complete problem is discussed in this paper. The simulation results shown in this paper proves that we are achieving efficient coverage with minimum number of sensors.

Keywords: sensor coverage, Memetic algorithm, set k cover problem, NP complete

I. INTRODUCTION

A wireless sensor network consists of a number of wireless sensor nodes. These nodes are characterized by being very small in size with limited energy usually supplied by a battery. Wireless sensor networks are a rapidly growing area for research and commercial development .Genetic Algorithms [1] are search algorithms based on the concepts of natural selection and natural evolution. Genetic Algorithm start with a set of solutions called initial population. Good initial population facilitates a GA's convergence to good solutions while poor initial population can hinder Genetic Algorithms (GA) convergence [2]. The random initialization is default method of population generation. Genetic algorithm uses an iterative process to create a population. The algorithm stops when the population converges towards the optimal solution. Hybridization of genetic algorithm with local search is known as memetic algorithm [3]. The theoretical foundations of genetic algorithm were originally developed by Holland [4].The idea of genetic algorithm based on the evaluation process of biological organism in nature. The basic steps of a simple Genetic Algorithm are show. A more comprehensive overview of GA Can be found in Refs [5,6,7,8].

Generate an initial population.
Evaluate fitness of individuals in the population.
Repeat
Select parents from the population;
Recombine parents to produce children;
Evaluate fitness of the children;
Replace some or of the population by the children;
Until a satisfactory solution has been found.

Memetic algorithm (MA) is a type of evolutionary algorithm. MA is invented by Dawkins in 1976, [9] inspired by memes, pieces of mental idea like stories, ideas and gossip, which reproduce (propagate) themselves through population of memes carriers. Hill climbing search was used to generate the initial population and then the performance was compared with Simple Genetic Algorithm (SGA)[10].

The objective of this paper is to:

- Analyze set k cover problem to enhance coverage.
- Analyze NP Complexity problems.
- Design Memetic algorithm for efficient coverage with minimum number of sensors.

II. ISSUES IN WIRELESS SENSOR NETWORK COVERAGE

A. *SET K COVER PROBLEM*

Slijepcevic et al [11] introduce the SET K-COVER problem identifying more sensor covers allows the WSN lifetime to be extended further. The problem of finding the maximum number of covers to extend WSN lifetime has been modeled as the SET K-COVER problem. Provided K covers, the lifetime of WSNs can ideally be extended by a factor of K using the above approach under the coverage constraint. The SET K-COVER problem has been proven to be NP-complete. Some heuristic algorithms have been presented, generally suffer from the trade-off between solution quality and running time. A means of improving solution quality with a short running time is urgently needed. This study develops a Memetic algorithm (MA) to solve the SET K-COVER problem of extending WSN lifetime. Suppose n sensors S1, . . . , Sn are deployed to monitor m targets T1, . . . , Tm. A target Tj is said to be covered if it lies within the sensing range of at least one sensor. Figure. 1 shows a WSN with five

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

sensors and four targets. The relationship between sensors S1, . . . , S5 and targets T1, . . . , T4 is represented by a bipartite graph G = (V,E), where V = S [T and eij 2 E if Si covers Tj. Figure. 2 presents the bipartite graph of the WSN in Figure. 1, where S1 = {T1}, S2 = {T1,T2}, S3 = {T2, T3, T4}, S4 = {T3}, and S5 = {T4}.

The maximum number K of disjoint covers in this example is two. They are C1 = {S1, S3} and C2 = {S2, S4, S}.The SET K-COVER problem is to find the maximum number of covers associated with the longest lifetime extension, which is equivalent to partitioning the set of sensors into the maximum number of covers. The problem is defined formally below.

Definition (SET K-COVER Problem) Given a collection S = {S1 . . . Sn} of subsets of a finite set T = {T1 . . . Tm}, find the maximum number, K, of covers C1 . . . CK # S with Ci \ Cj = ; for i – j, such that every element of T belongs to at least one element of Ci.

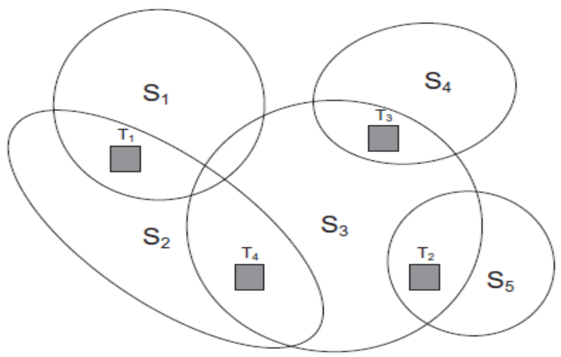

Figure 1: Example deployment of WSN

B. NP-complete

In computational complexity theory, the complexity class NP-complete (abbreviated NP-C or NPC) is a class of decision problems. A decision problem L is NP-complete if it is in the set of NP problems so that any given solution to the decision problem can be verified in polynomial time, and also in the set of NP-hard problems so that any NP problem can be converted into L by a transformation of the inputs in polynomial time.

Although any given solution to such a problem can be verified quickly, there is no known efficient way to locate a solution in the first place; indeed, the most notable characteristic of NP-complete problems is that no fast solution to them is known.

That is, the time required to solve the problem using any currently known algorithm increases very quickly as the size of the problem grows. As a result, the time required to solve even moderately large versions of many of these problems easily reaches into the billions or trillions of years, using any amount of computing power available today.

As a consequence, determining whether or not it is possible to solve these problems quickly, called the P versus NP problem, is one of the principal unsolved problems in computer science today as shown in figure 2.

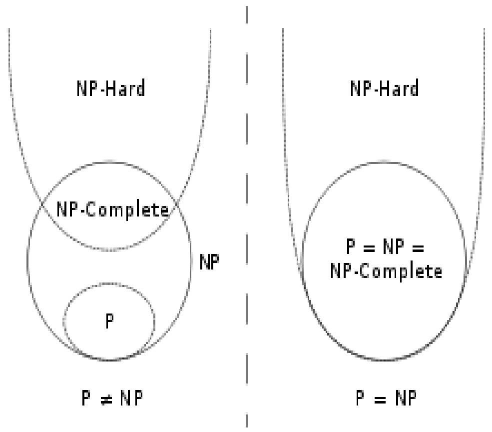

Figure 2: Euler diagram for P, NP, NP-complete, and NP-hard set of problems.

While a method for computing the solutions to NP-complete problems using a reasonable amount of time remains undiscovered, computer scientists and programmers still frequently encounter NP-complete problems. NP-complete problems are often addressed by using approximation algorithms.

C. NP (complexity)

In computational complexity theory, NP is one of the most fundamental complexity classes. The abbreviation NP refers to "non-deterministic polynomial time"

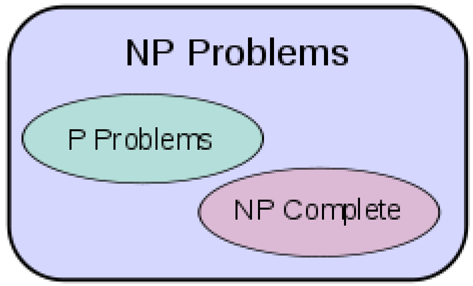

Figure 3: Diagram of complexity classes provided that P ≠ NP. The existence of problems outside both P and NP-complete in this case was established by Ladner.

Intuitively, NP is the set of all decision problems for which the instances where the answer is "yes" have efficiently verifiable proofs of the fact that the answer is indeed "yes". More precisely, these proofs have to be verifiable in polynomial time by a deterministic Turing machine. In an equivalent formal definition, NP is the set of decision problems where the "yes"-instances can be recognized in polynomial time by a non-deterministic Turing machine. The equivalence of the two definitions follows from the fact that an algorithm on such a non-deterministic machine consists of two phases, the first of which consists of a guess about the solution which is generated in a non-deterministic way, while the second consists of a deterministic algorithm which verifies or rejects the guess as a valid solution to the problem.

The complexity class P is contained in NP, but NP contains many important problems as shown in figure 3,

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

the hardest of which are called NP-complete problems, for which no polynomial-time algorithms are known. The most important open question in complexity theory, the P = NP problem, asks whether such algorithms actually exist for NP-complete, and by corollary, all NP problems. It is widely believed that this is not the case.

III. DESIGN AND IMPLEMENTATION MEMETIC ALGORITHM.

Memetic algorithms (MA) represent one of the recent growing areas of research in evolutionary computation. The term MA is now widely used as a synergy of evolutionary or any population-based approach with separate individual learning or local improvement procedures for problem search. Quite often, MA are also referred to in the literature as Baldwin EAs, Lamarckian EAs, cultural algorithms or genetic local search. History Inspired by both Darwinian principles of natural evolution and Dawkins' [5] notion of a meme, the term "Memetic Algorithm" (MA) was first introduced by Moscato in his technical report in 1989 where he viewed MA as being close to a form of population-based hybrid genetic algorithm (GA) coupled with an individual learning procedure capable of performing local refinements. The metaphorical parallels, on the one hand, to Darwinian evolution and, on the other hand, between memes and domain specific (local search) heuristics are captured within Memetic algorithms thus rendering a methodology that balances well between generality and problem specificity. In a more diverse context, Memetic algorithms are now used under various names including Hybrid Evolutionary Algorithms, Baldwinian Evolutionary Algorithms, Lamarckian Evolutionary Algorithms, Cultural Algorithms or Genetic Local Search. In the context of complex optimization, many different instantiations of Memetic algorithms have been reported across a wide range of application domains, in general, converging to high quality solutions more efficiently than their conventional evolutionary counterparts. The combination of Evolutionary Algorithms with Local Search Operators that work within the EA loop has been termed "Memetic Algorithms" and this MA have been shown to be orders of magnitude faster and more accurate , and are the "state of the art" on many problems. The concept of meme is being implemented in Memetic Algorithms to solve optimization problems. In order to do so, the term meme is firstly defined and then compared with currently implemented memes in Memetic Algorithms. Thus, we are able to determine how far the concept of meme is being used, or in fact if it is being used at all. Modifications to the algorithm to further mimic the concept of meme are then suggested. A meme is defined to be 'a noun that conveys the idea of a unit of cultural transmission, or a unit of imitation' (Dawkins). In other words, a meme can be considered as any unit of information, observable in the environment. They are similar to genes in that they are self-replicating, but memes differ from genes in that they are transmitted through imitation rather than being inherited. Furthermore, memes replicate in a Lamarckian manner (rather than in a Darwinian manner) in that changes to during its lifetime are passed on. Examples of memes are catch phrases, stories, fashion, technology and chain letters. A population-based search algorithm called Genetic Algorithm (GA) is commonly used to solve combinatorial optimization problems where the goal is to find the best solution in a (possibly unknown) solution space. It uses the principle of biological evolution to generate successively better solutions from previous generations of solutions. Memetic algorithm (MA) is an extension of GA which incorporates a local-search algorithm for each solution in between generations. According to Pastorino, MA is able to improve convergence time, hence making it more favorable over GA. Figure 4 shows the scheme for a typical MA.

It is evident that local search is performed in between each generation, in addition to the techniques used by GA to explorethe search space, namely recombination/crossover and mutation. For this reason, Memetic Algorithm is also known as Hybrid-GA. Local search is performed to improve the fitness of the population (in a localized region of the solution space) so that the next generation has "better" genes from its parents, hence the claim that Memetic Algorithms can reduce convergence time. Memetic Algorithms incorporate the concept of memes by allowing individuals to "change" before the next population is produced.

Individuals may "copy" parts of genes from other individuals to improve their own fitness. The local search algorithm adopted in a Memetic Algorithm is somewhat dependent on the problem being solved; however the common trait with any local search is that parameters in the algorithm cannot be changed.

This does not follow with the definition of a meme, in that it can be changed because the adopted meme is the individual's own interpretation of it. Furthermore, when memes are transmitted, changes to them are also passed on. Memes affect the behavior of an individual, and do not modify the genes themselves. However, as a practical issue, a meme in a Memetic Algorithm must be able to modify genes in order to improve fitness during local search.

The Pseudo codes of the standard MA is as follows:

BEGIN
Initialize: *Generate a database containing a population of designs.*
While *(computational budget is not exhausted) Evaluate all individuals in the population using the exact fitness function.*
For *each non-duplicated individual in the EA population*
MA: *Apply local search strategy using original exact fitness functions.*
End for
Replace the individuals in the population with the locally improved solution
Apply standard EA operators to create a new population.
End While
END

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

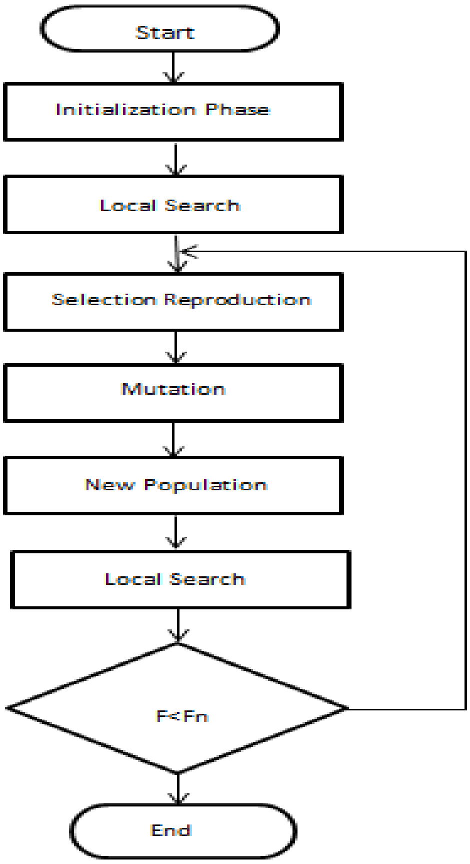

Figure 4: Scheme of the Memetic algorithm.

To further incorporate the concept of memes into MAs, local searches could involve a history dependent component. This history component can be referred to as a trend, whereby it could dictate or influence the outcome of the local search. A trend could be defined as a particular search direction, or a particular gene(s) to imitate. The local search algorithm is applied to wireless sensor networks are shown in figure 5. Like current local searches, trends would be dependent on the problem being solved.

Each individual in the population has a probability of adopting this trend, or a typical local search algorithm. If the trend is not adopted and the individual finds a better fitness on its own, it may change the trend. The best fit method is used for better solution as shown in figure 6. This trend could be reinforced as more individuals adopt it (i.e. more chance of following the trend), but reduces in popularity if following this trend results in a lowered fitness. Compared to before, this trend is not inherited through recombination, but exists throughout each generation to the end. This trend could change several times during iterations, and affect individuals in different ways. This is very similar to trends or fashions in human society, where they appear and disappear or change over a period of time.

Memetic algorithm is a blooming dialect of evolutionary algorithm (EA). MA adopts the Lamarc kian theory that offspring can inherit the knowledge or characteristics that their parents acquire during their lifetime. The MA implements this idea by integrating a local enhancement, such as local search and repair operator, into the canonical EA, and making the enhancement inheritable. This integration significantly improves the exploitation ability of EA and has been widely shown to provide superior solution quality and high convergence speed .The proposed MA is based on the order-based GA with the compact operator, which is a novel local enhancement operator that is designed to address the SET K-COVER problem. Furthermore, this study devises a fitness function based on the contribution of sensors to covers as shown in figure 7.

A series of simulations is conducted to evaluate the performance of the proposed MA in terms of solution quality and running time, and to verify its superiority over several heuristic and evolutionary algorithms.

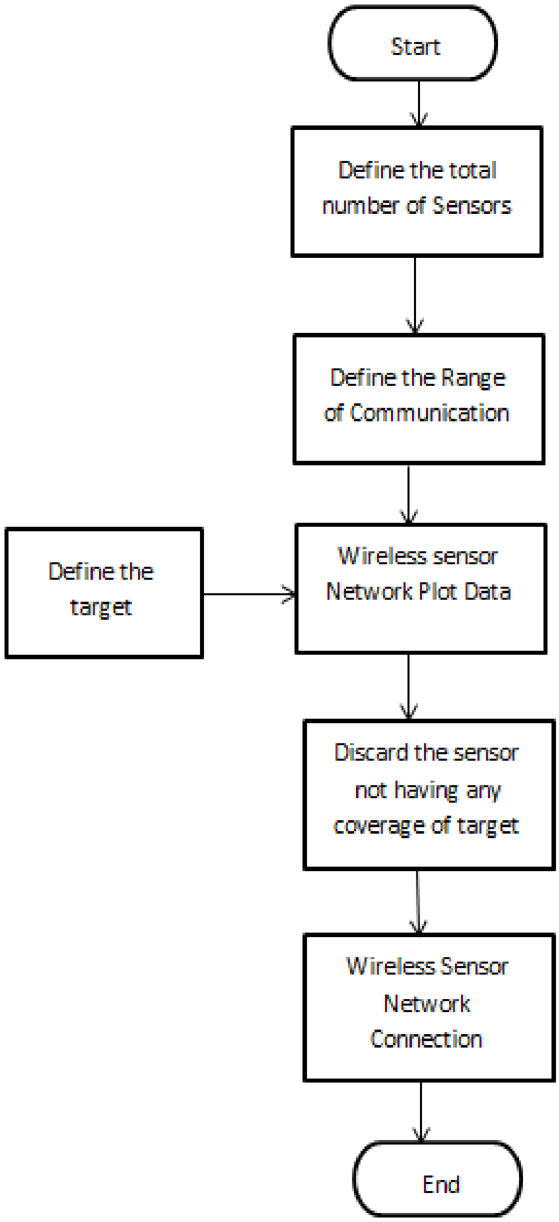

Figure 5: The local search algorithm is applied to wireless sensor networks

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

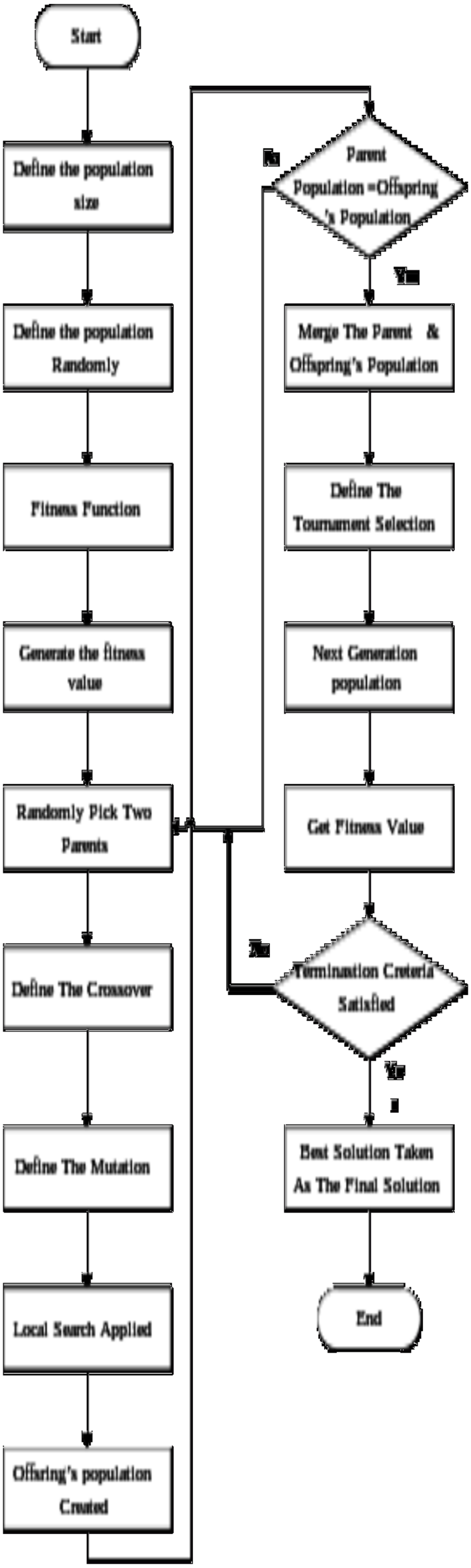

Figure 6: Best fit methods to obtain better solution

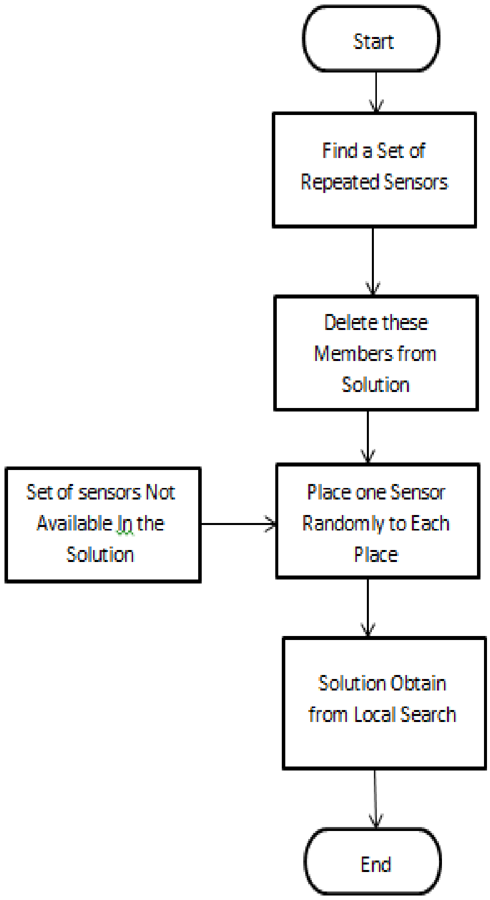

Figure 7: Order-based GA with the compact operator, which is a novel local enhancement operator that is designed to address the SET K-COVER problem.

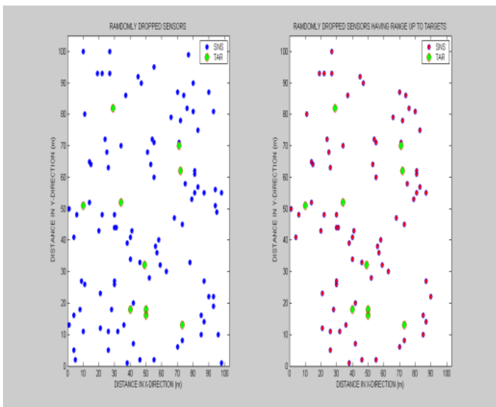

Figure 8: sensors are deployed

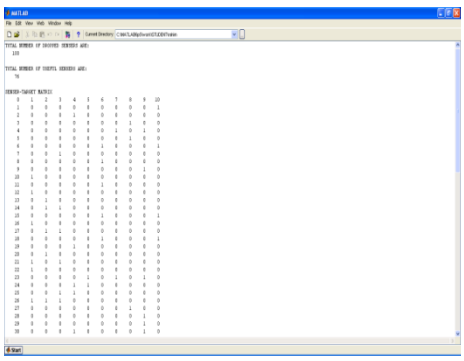

Figure 9: simulation results before applying the algorithm

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

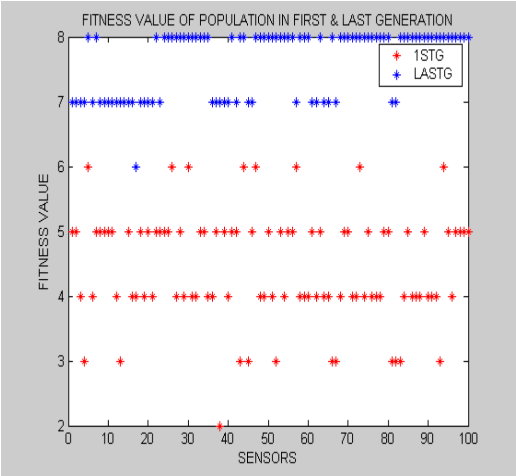

Figure 10: simulation results by using best population search

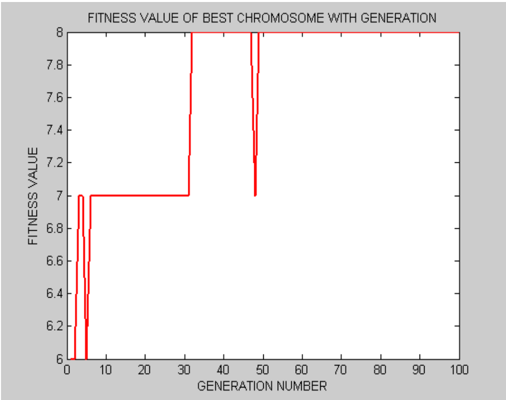

Figure 11: simulation results for fitness value

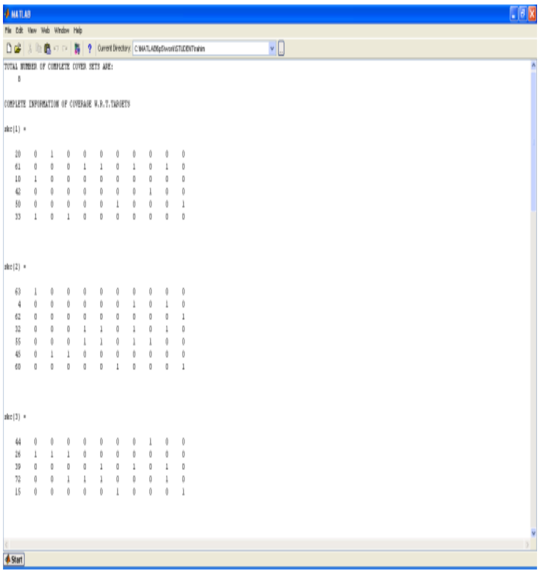

Figure 12: simulation results after applying the algorithm

IV. CONCLUSION

This paper deals with the sensor coverage problem is prime factor being considered in recent advances. We analyze set k cover coverage method to minimize the number of sensors with high coverage and NP complete problem analysis for different kind of sensor typologies. We are design and implemented Memetic algorithm for efficient coverage in wireless sensor with minimum number of sensor using best population fit algorithms. The simulation results shown the numbers of sensors are deployed before applying the Memetic algorithm and after applying the MA .The results are prove that number of sensors required is less in number with efficient coverage.

REFERENCES

[1] D.E. Goldberg, Genetic algorithms in search, optimization, and machine learning. Addison Wesley Longman, Inc., ISBN 0-201-15767-5, 1989.

[2] Raymond R. Hill (1999), "A Monte Carlo Study of genetic algorithm initial population Generation method", In Proceeding of the 1999 Winter simulation Conference, on Genetic Algorithms, Lawrence Erlbaum Associates, pp 1-8.

[3] Swapna Devi, Devidas G. Jadhav ,Shyan S.Pattnaik, "Memetic algorithm and its application to function optimization and noise removal"

[4] J.H.Holland, adaption in nature and artificial system, MIT press, cambrige MA, 1975.

[5] Beasely, D.R. Bull, and R.R.Martin "AN overview of genetic algorithms: Part I Reaserch topics" University Computing 15(1993) 58-69.

[6] Beasely, D.R. Bull, and R.R.Martin "AN overview of genetic algorithms: Part II Reaserch topics" University Computing 15(1993) 170-181.

[7] D.Golberg, Genetic Algorithms in search, Optimization and machine learning, Addison- Wesley, New York 1989.

[8] C.R.Reeves, Modern Heuristic Techniques for combinatorial problems, Blackwell Scientific , 1993

[9] R. Dawkins, The Selfish Gene, Oxford University Press, Oxford, UK, 1976.

[10] S.Padmavathi, S.Mercy Shalinie and R.Abhilaash , "A Memetic Algorithm Based Task Scheduling considering Communication Cost on Cluster of Workstations" Int. J. Advance. Soft Comput. Appl., Vol. 2, No. 2, July 2010

[11] S. Slijepcevic and M. Potkonjak, "Power efficient organization of wireless sensor networks," in IEEE International Conference on Communications, vol. 2, 2001, pp. 472–476.

BIOGRAPHIES

Mr. Channakrishnaraju has 17 years of teaching experience for UG and PG courses in computer Science and Engg and presently working as Associate Professor in the department of computer Science and Engg at Sri Siddhartha Institute of Technology, Tumkur. He Obtained B.E from Bangalore University in the year 1995 and PG in software systems in the year 2000. His research interests are in the areas Wireless sensor networks and Artificial Intelligence. Currently pursuing doctoral degree in Computer Science &Engg (wireless sensor networks) from Sri Siddhartha Academy of Higher Education [SSAHE], Tumkur, under the guidance of Dr. M.Siddappa Professor and Head of the department of Computer Science and Engineering, Sri Siddhartha Institute of Technology, Tumkur. He published 17 Technical Papers in National, International Conference and journals.

M.Siddappa received B.E degree in Computer Science & Engineering from University of Mysore, Mysore, Karnataka, India in 1989, M.Tech from University of Mysore in 1993 and doctoral degree from Dr .MGR Educational Research Institute Chennai under supervision of Dr.A.S.Manjunatha, CEO, Manvish e-Tech Pvt. Ltd., Bangalore. He worked as project associate in IISc, Bangalore under Dr. M.P Srinivasan and Dr. V.Rajaraman from 1991 – 1993. He has teaching experience of 22 years and research of 5 years. He published 35 Technical Papers in National, International Conference and Journals. He is a member of IEEE and Life member of ISTE. He is working in the field of data structure and algorithms, Artificial Intelligence, Image processing and Computer networking. He worked as Assistant Professor in Department of Computer Science & Engineering from 1996 to 2003 in Sri Siddhartha Institute of Technology, Tumkur. Presently, he is working as Professor and Head, Department of Computer Science & Engineering from 1999 in Sri Siddhartha Institute of Technology, Tumkur. He has published nearly 33 papers in national and international journals. He has received best Engineering teacher award from STE for the year 2011.