Int J Wireless Inf Networks (2016) 23:246–256
DOI 10.1007/s10776-016-0317-0

Maximization of WSN Life Using Hybrid Evolutionary Programming

P. Nagarathna[1] · R. Manjula[1]

Received: 24 February 2016 / Accepted: 19 July 2016 / Published online: 28 July 2016

Abstract Of all the challenges faced by wireless sensor networks (WSN), extending the lifetime of the network has received the most attention from researchers. This issue is critically important, especially when sensors are deployed to areas where it is practically impossible to charge their batteries, which are their only sources of power. Besides the development and deployment of ultra low-power devices, one effective computational approach is to partition the collection of sensors into several disjoint covers, so that each cover includes all targets, and then, activate the sensors of each cover one at a time.. This maximizes the possible disjoint covers with an available number of sensors and can be treated as a set-K cover problem, which has been proven to be NP-complete. Evolutionary programming is a very powerful algorithm that uses mutation as the primary operator for evolution. Hence, mutation defines the quality and time consumed in the final solution computation. We have applied the self adaptive mutation strategy based on hybridization of Gaussian and Cauchy distributions to develop to develop a faster and better solution. One of the limitations associated with the evolutionary process is that it requires definition of the redundancy covers, and therefore, it is difficult to obtain the upper bound of a cover. To solve this problem, a redundancy removal operator that forces the evolution process to find a solution without redundancy is introduced. Through simulations, it is shown that the proposed method maximizes the lifespan of WSNs.

Keywords Wireless sensor network · Life enhancement · Evolutionary programming · Gaussian mutation · Cauchy mutation

1 Introduction

Rapid development in the field of micro-electro-mechanical and microelectronics in the past few decades has influenced a lot of applications that have improved peoples lives. A WSN is one such technology that has influenced many applications readily used in research, health monitoring and control, simple home appliances, and by the military. A WSN is made up of small-sized multifunctional sensor nodes that communicate over short distances. These sensor nodes are equipped with sensing, data processing, and communicating capabilities. The on-board processor in sensor nodes is responsible for taking raw data as input from its environment, performing simple processing of raw data and transmitting only desired partially processed data. Sensor nodes are equipped with on board batteries with very limited energy and recharging or replacing these batteries that have very limited energy, and replacing and recharging these batteries is impossible to do when sensors are used in certain environments like erupting volcanoes, nuclear reactors and areas that have experienced disasters. Since it is very difficult to change batteries of sensor nodes, it is very important to prolong the lifetime of sensor nodes for as long as possible once the sensors are deployed.

A large number of sensors are densely deployed in and around the targets to form a WSN that monitors a finite number of targets in its range [9, 10]. The random

✉ R. Manjula
rmanjula@vit.ac.in

P. Nagarathna
pnagaratna@gmail.com

[1] School of Computer Science and Engineering (SCOPE), VIT University, Vellore 632014, India

distribution and deployment is possible as there is no constraint on the positioning of sensors, and this provides access to inaccessible terrains or areas that are the focus of disaster relief operations.

On the ot her hand, sensor network protocols and algorithms must possess self-organizing capabilities, and cooperative effort is expected from sensor nodes to achieve the required functionality [14, 23]. If the network supports all these features, then it ensures a wide range of applications. It also provides the end user with enough intelligence and a better understanding of the environment that is being monitored. In the near future, wireless sensor networks may become an integral part of our lives, serving many purposes in the same manner as the present-day computers and mobiles.

The wireless sensor node is a microelectronic device that has a limited power source, and designers of networks must ascertain that it works for long durations once it is established. In other words, a WSNs lifetime depends on each sensor nodes lifetime or the batterys lifetime. Therefore, if the battery of each sensor node is efficiently utilized then it increases the lifetime of a WSN. Hence, power conservation and power management is of at most importance and requires significant attention while addressing this problem. Many researchers nowadays are focusing on the design of power-aware protocols and algorithms for sensor networks because of the above mentioned importance.

In this paper, we propose algorithms for finding the maximum number of disjoint set-K covers that can enhance the lifetime of a wireless sensor network. The paper is organized as follows: Section 2 discusses related studies that have explored maximization of the lifetime of WSNs; Sects. 3 and 4 discuss disjoint set cover problems and evolutionary programming respectively; Sect. 5 discusses the proposed model for enhancing the lifetime of WSN; Sect. 6 deals with result analysis and inference from the proposed model; and Sect. 7 deals with the conclusion and future direction in the similar line of study.

2 Related Work

Kumar and Chatuverdi [12] proposed two algorithms called *k*-means and fuzzy c-means for network routing under severe energy constraint situations in WSNs. Here, the total sensor nodes are divided into several clusters, and the cluster head is chosen based on the residual energy status of sensors and Euclidean distances. Anitha and Kamalakkannan [3] proposed an have proposed an algorithm called low energy adaptive clustering hierarchy—mobile (LEACH-M), which is an enhanced cluster-based algorithm for optimal routing in WSNs. Chen et al. [6] proposed a routing algorithm for WSNs that uses neural networks, and here, the cluster head is selected after a result of competitive learning among nodes. In this algorithm, the residual energy of the sensors and distances among neighbor nodes are considered for cluster processing.

Multimedia messages are very common over the internet and, in turn, WSNs. Studies have explored a wireless personal area network that supports many multimedia applications which are low cost and support low-power operations [21]. Berger-Wolf et al. [5] used sensor network to detect and identify the source of contamination with two variants to address this problem namely sensor constained and time constrained. They also showed that both variant problems are polynomially equivalent. Shu et al. [20] used Fuzzy logic system to optimally distribute the randomly placed sensors. They also showed that their approach provides faster and energy efficient sensing of the area of interest or the area to be sensed.

Halder et al. [11] proposed an energy conservation algorithm that avoids energy holes and enhances WSN lifetime. Others have proposed scheduling the sensor nodes into disjoint set-K covers to extend the lifetime of WSNs [13], using a selfstabilizing algorithm in an efficient optimization technique [4] and utilizing a coverage algorithm that creates both disjoint and non-disjoint set-K covers, and improves network functionality by using a cost function that checks the monitoring capabilities of a sensor node and poorly monitored targets [25]. Dietrich and Dressler [8] presented a survey on WSNs lifetimes using a simulation model.

Alshawi et al. [2] proposed a routing algorithm for increasing the lifetime of WSNs that was based on fuzzy logic and star algorithms. An algorithm based on duty cycle [18] and a method that uses the residual battery energy of sensor nodes to adaptively adjust their transmission range have been proposed to increase the lifetime of WSNs, as well. Wang et al. [22] proposed a mechanism that works in both the physical layers and the MAC of data link layers of WSNs to increase their lifetimes. They formulated the problem of lifetime extension as a mixed integer convex optimization problem and used time division multiple access (TDMA). Other studies exploited the advantages of the imperialist competitive algorithm(ICA) to determine set-K covers [24].

Cheng and Yang [7] present an opportunistic reception (OR) algorithm for achieving energy efficiency in WSNs. Here number of packets sent and received by node(s) is opportunistically reduced by forcing the intermediate cluster to generate independent coding vectors through simple decoding and recoding. Mehra and Dabas [15] proposed energy efficiency. Energy efficiency in WSNs improves their lifetime, and a method that uses this approach based on the SPIN-I protocol was proposed by Pattani and Chauhan [17]. A metaheuristic population based soft computing algorithm has been proposed has been proposed as an optimization technique for extending the lifetimes of WSNs [19]. It uses a bat swarm optimization technique to reduce intra-cluster compactness and, thus, reduce energy consumption.

Extending WSNs lifetime by disjoint set-K cover was first proposed by Abrams et al. [1]. Padmavathy and Chitra [16] centralized scheduling algorithm for forming disjoint or non-disjoint K covers to maximize the lifetime of WSNs. Network performance was also improved by turning on redundant nodes only when required, otherwise they were in an inactive mode.

3 Disjoint Set Cover (DSC) Problem

The main goal of a WSN is to cover all targets with a minimum number of sensors, which means a network must use few sensors to cover all targets at any one time. In practice, a large number of sensors are randomly dropped, which may result in more than one sensor covering the same target. In turn, this may result in collection of redundant information about a single target by many sensors, and moreover, many sensors waste their limited power collecting information that might have already been collected by other sensors.

To increase the life span of sensor network, many disjoint subsets of sensors are created in such way that each subset covers all targets. Therefore, when one subset is activated, it will cover all targets, providing functionality to the WSN.

However, in order for the sensors to have a maximum lifetime, they should have a maximum number of disjoint subsets, and forming the maximum number of disjoint subsets from the available sensors is considered as disjoint set cover problem. Technically, it can be described as follows: ”if there are T number of targets which has to be covered by S number of sensors, then divide the S sensors into K different sets in such a manner that there is no common sensors in any set and each set can meet all the objectives by itself”.

4 Evolutionary Programming

Evolutionary computation is part of the natural computing intelligence that is applied to various optimization problems. In evolutionary computation, a population progresses with time through an iterative process. When we use evolutionary computation to solve a problem, we may provide guidelines that dictate how change must take place in each iteration so that a new and improved solution is obtained with each iteration.

During the evolutionary process, evolution is achieved through random change and selection of the fitness function. Initially, a population called the parent population is defined randomly. Members of population creates the offspring either at the individual level or in combination form by the random variation operator through crossover, mutation or both. This random variation operator basically works as an exploration in the solution computation phase, and it is used to create a new solution in the solution space. It is also responsible for maintaining the diversity of the population.

Different types of distributions can be considered in the development of the random search operator, which is also considered to be the offspring creator. In order to assume that the population for the next generation is limited, it is necessary to apply some type of selection criteria. Fitness becomes the primary parameter for selection of a solution, but care needs to be taken to make sure the selection process is unbiased, otherwise there may be a loss of exploitation of a search process and evolution may not work properly. Termination may be forced, where by a limited number of iterations are allowed (but there is no guarantee of optimal evolution) where as in case of self termination process, or or it may be self-determined, where by the evolution itself decides when to stop. Generally, when there is no improvement in fitness with a number of iterations, then the evolution may be considered to be terminated.

The procedure for evolutionary process to create the new solution can be defined mathematically by Eq. 1

$$X(t+1) = fs(r(X[t])) \tag{1}$$

where X[t] is taken as the population at time t, fs is the selection operator and r is a random variation operator. A number of possibilities can define the variation and selection functions that play a role in the quality of evolution. Different methods of evolutionary computation exist, like evolutionary programming, evolution strategies and genetic algorithms, where the nature of evolution is different under same evolution structure.

In genetic algorithms, crossover is the primary operator for random variation and mutation is the second, whereas in evolutionary programming and strategy, mutation is the key operator for random variation. Some differences also occur in the representation of the solution or chromosome in a population: either it is coded in some form, or it may take a real value of a parameter that is available in the solution.

4.1 Mutations and Distributions

In the heuristic method, the fundamental principles of exploration and exploitation play a central role. With better exploration, there is a good chance of getting the better solution, and with proper exploitation, the best solution can be selected from the explored solutions; however, when exploration is emphasized the program takes a long time to converge, and it converges prematurely when exploitation is emphasized. The balance between these two operations decides the quality of the solution obtained at the end. In the proposed solution, the central heuristic operator, the mutation operator, is a hybrid of the Gaussian and Cauchy probability distributions.

In evolutionary programming, mutation plays a primary role in the creation of offspring. The quality of the mutation mechanism will decide the generated offsprings diversity and genetic inheritance. It is obvious that the balance between genetic inheritance and defined diversity will decide the quality of exploration. Mathematically, the mean centric distribution is used for defining the mutation. For example, Gaussian distribution (with Mean 0 and Standard Deviation of 1) is the traditional choice for the mutation mechanism; however, its large variation has low probabilities that prevent a quick exploration of solutions. To overcome this issue one can consider Cauchy distribution as an optimal choice since it is mean centric and has a longer tail.

As shown in Fig. 1, the Gaussian distribution has a shorter tail that makes the probability values low for high numbers on either side of the mean value. In comparison, the Cauchy distribution, whose shape is similar to that of the Gaussian distribution, has a longer tail that provides a better chance for obtaining high probability values on both sides of the mean value.

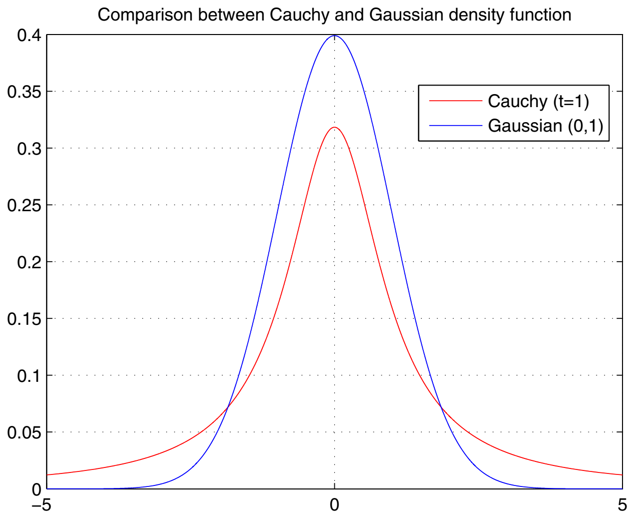


Fig. 1 Cauchy and Gaussian density function

4.2 Cauchy and Gaussian Distribution

The one dimensional Cauchy density function centered at the origin with $t = 1$ is defined by equation 2

$$f_1(x) = \frac{1}{\pi(1+x^2)}, \qquad -\infty < x < \infty \tag{2}$$

The corresponding distribution function is defined by equation 3

$$F_1(x) = \frac{1}{2} + \frac{1}{\pi}(\arctan(x)) \tag{3}$$

Similarly, the Gaussian density function with Mean $(\mu) = 0$ and Standard Deviation $(\sigma) = 1$ is given by Eq. 4

$$f_2(x) = \frac{e^{\frac{-x^2}{2}}}{\sqrt{2\pi}}, \quad \text{for all } x \in \mathbb{R} \tag{4}$$

The shape of $f_1(x)$ resembles that of the Gaussian density function but approaches the axis so slowly that an expectation does not exist. As a result, the variance of the Cauchy distribution is infinite. The comparison of Cauchy and Gaussian density function is shown in Fig. 1.

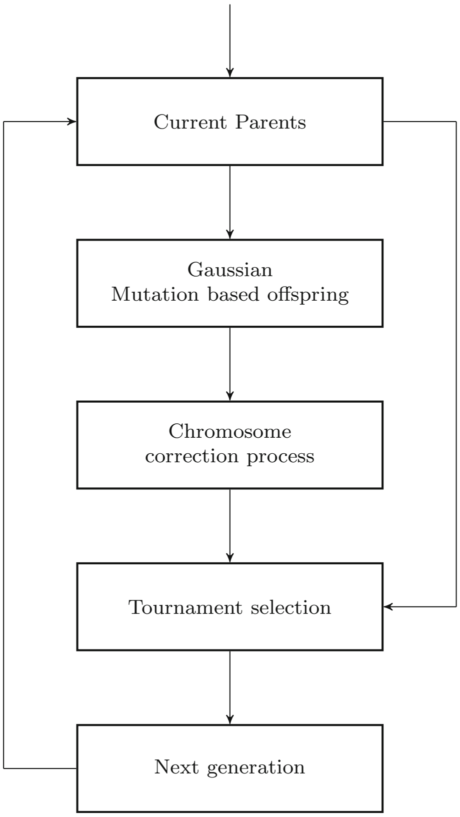

Current Parents
Gaussian Mutation based offspring
Chromosome correction process
Tournament selection
Next generation

Fig. 2 EVGM

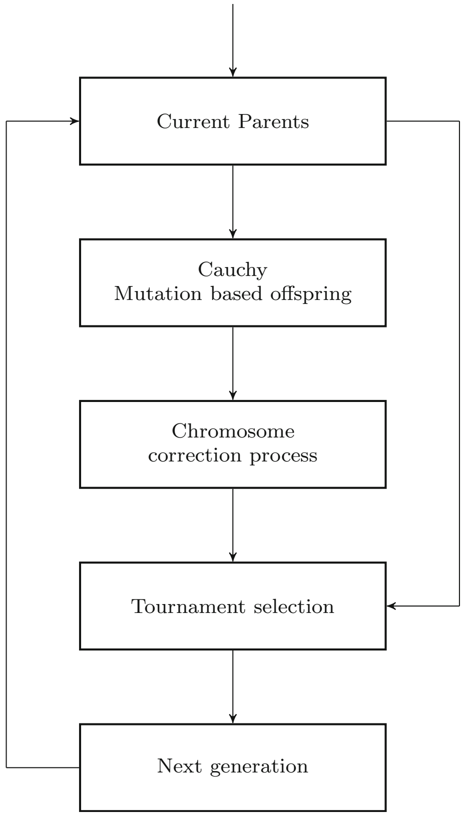

Current Parents
Cauchy Mutation based offspring
Chromosome correction process
Tournament selection
Next generation

Fig. 3 EVCM

5 Proposed Model for Life Enhancement of WSN with Evolutionary Programming

It is possible to know the upper bound of disjoint cover since it is equal to the minimum number(say K) of sensors which track a target (T_x), and there are more than K number of sensors available to track all the other targets. The upper bound of the disjoint cover can be used as a reference in designing the solution and conducting comparative analyses. We have used the phenotypic representation of chromosomes, which we have represented as sensor numbers. The fitness function is a very important part of evolutionary computation because the evolutions of solutions occur according to the fitness functions fitness judgment. Essentially, the description of the fitness function is problem-dependent and defined over the objectives. To enhance the lifetime of a WSN, the objective is to have a maximum number of covers so that in the fitness function, we can take the maximum number of disjoint covers available in a chromosome. Here, estimation of the covers started from the first sensor and sequentially progressed to the last sensor.Because of the mutation and rounding effects, some sensors in the chromosome may be repeated or not considered. To overcome this issue, a two-step correction process is applied. First, repeated sensors are identified and removed, and their positions remembered as

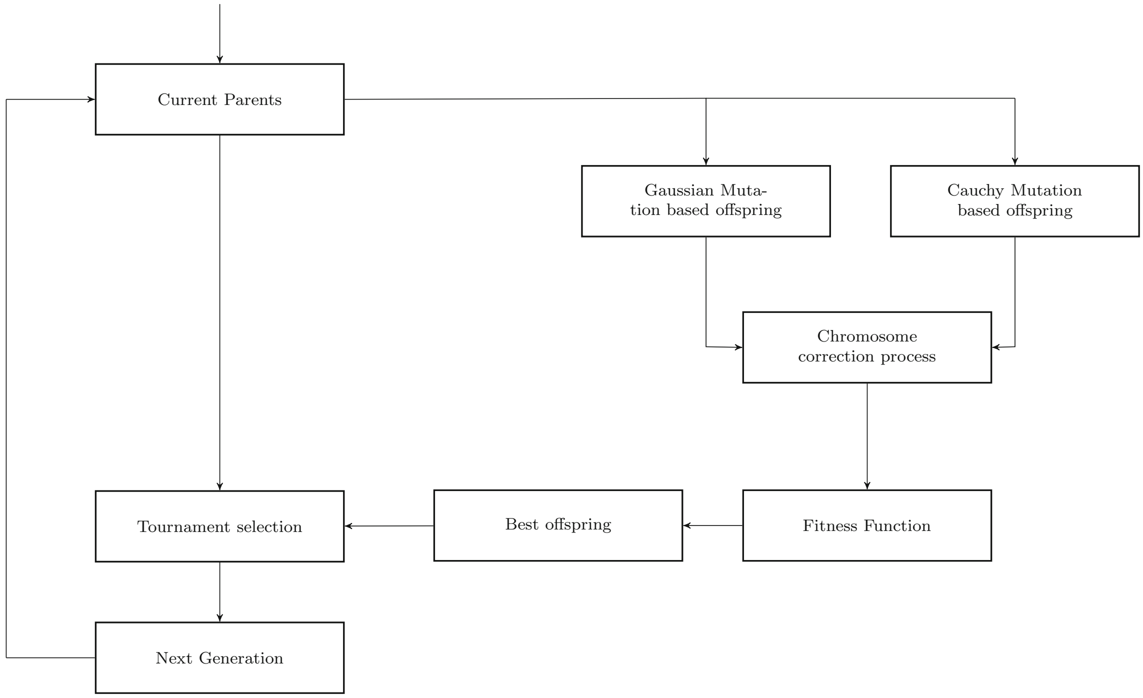

Current Parents
Gaussian Mutation based offspring
Cauchy Mutation based offspring
Chromosome correction process
Tournament selection
Best offspring
Fitness Function
Next Generation

Fig. 4 EVHM

places of vacancy. In the second step, sensors that were not part of the earlier solution are randomly allocated to the vacant positions. We have used four different sets of algorithms to develop the disjoint covers: (1) evolutionary programming with Gaussian mutation (EPGM); (2) evolutionary programming with Cauchy mutation (EPCM); (3) evolutionary programming with winner hybrid mutation (EPHM); and (4) evolutionary programming with winner hybrid mutation with redundancy reduction (EPHMRR). The functional block diagrams for all the algorithms are given in Figs. 2, 3, 4 and 5. The algorithmic approach of the evolutionary computation is described in Sect. 5.1.

We conducted a series of simulations to estimate the set-K cover under two different conditions. (Note that a sensor

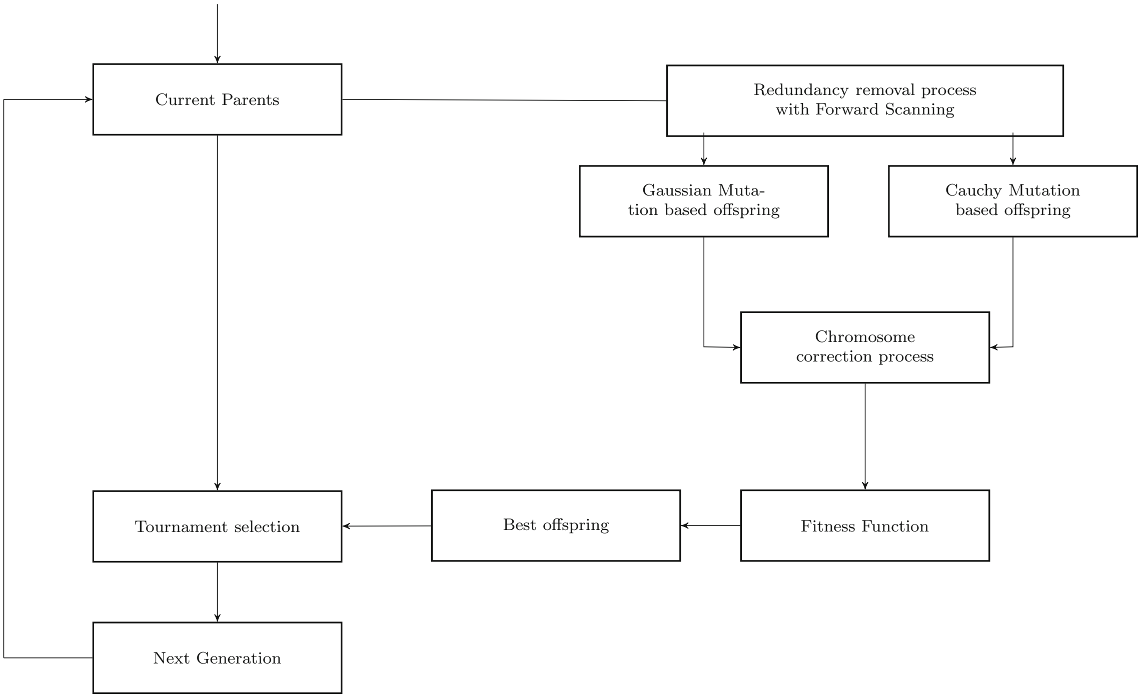


Fig. 5 EVMHHR

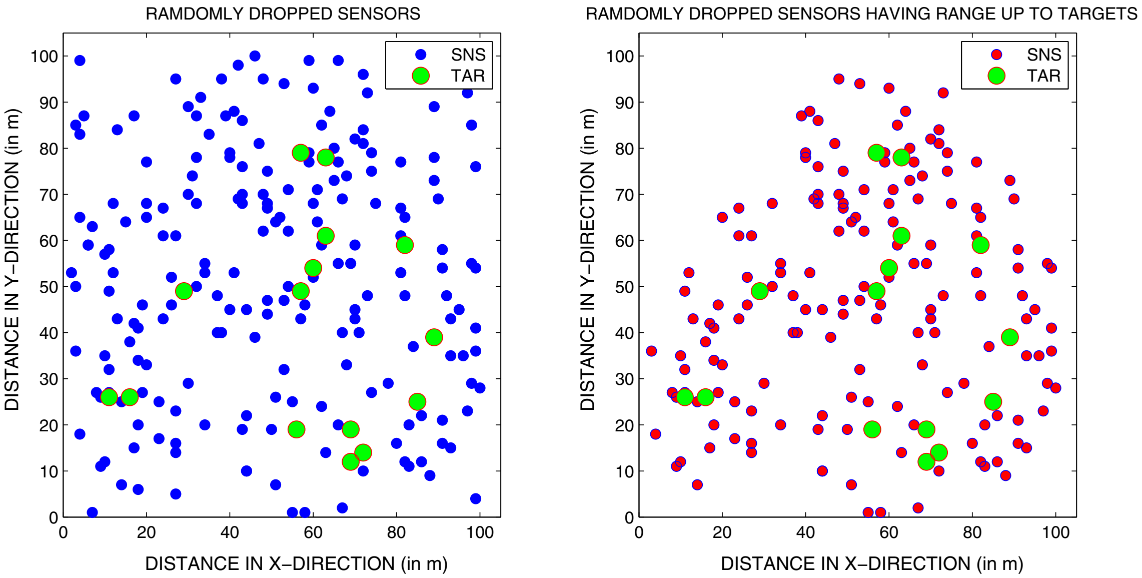


Fig. 6 WSN simulation development for Case 1

Table 1 Performance of different algorithms for Case 1

Trail no.	EVGM		EVCH		EVHM		EVHMRR	
	DC	RS	DC	RS	DC	RS	DC	RS
1	36	1	35	1	36	4	37	3
2	36	0	36	0	37	2	37	1
3	36	0	36	0	36	0	37	3
4	36	1	36	1	36	1	37	1
5	36	0	36	0	36	0	37	5

network is usually deployed at random.) In the first case, as shown in Fig. 6, we used 15 randomly defined targets, 200 sensors and a sensing range of 20 m, while in the second case, which is shown in Fig. 9, we used 10 randomly defined targets, 100 sensors and a sensing range of 50 m. The simulation platform was MATLAB 7.6.0 in a Windows XP operating environment. In all cases, the k cover was estimated with different solutions for 5 independent trails. The region of the experiment was defined as a 100 × 100 unit, and the solution was based on the evolution of 100 generations. In all cases, the sensor dropping distribution was assumed to be a uniform random distribution. In Table 1, the performance of all the algorithms is tabulated and compared for the first case, which had an upper bound value for the disjoint cover as 37. It is clear, from the result, that both EPGM and EPCM could not achieve the upper bound in any of the tails. EPHM showed a comparative advantage, achieving the upper bound in two trails, and EPHMRR clearly outperformed the other algorithms, achieving an upper bound in all trails. EPHMRR, also, left extra unused nodes that can be utilized in the future. Convergence towards a final solution for all trails is shown in Fig. 7. It can be observed that EPGM and EPCM have more fluctuations in solution development in comparison of EVHM and EVHMRR. The mean convergence characteristics of all the defined algorithms are given in Fig. 8. Tables 2 and 3 show comparisons of the number of K covers obtained in case 2 under two population sets. The corresponding convergence characteristics are given in Figs. 10, 11 respectively. It was difficult to obtain the

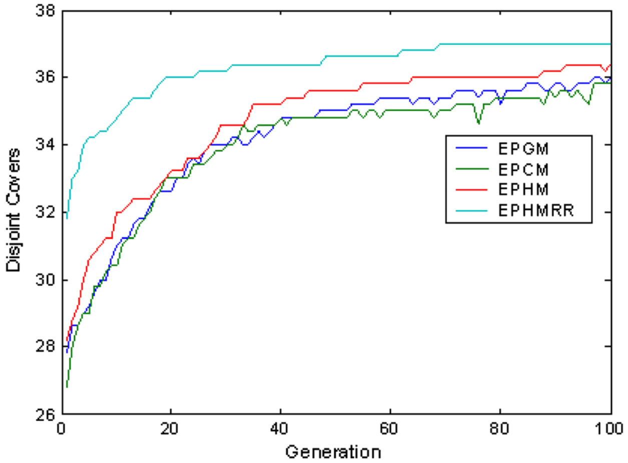
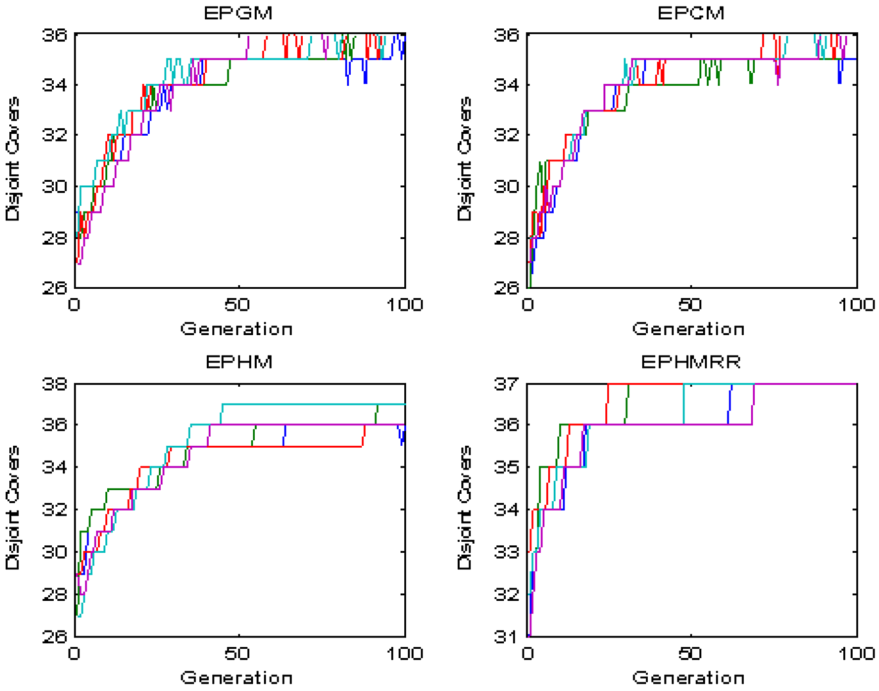


Fig. 8 Mean convergence characteristics of different algorithms for Case 1

Fig. 7 Solution convergence characteristics of different algorithms for five independent trails



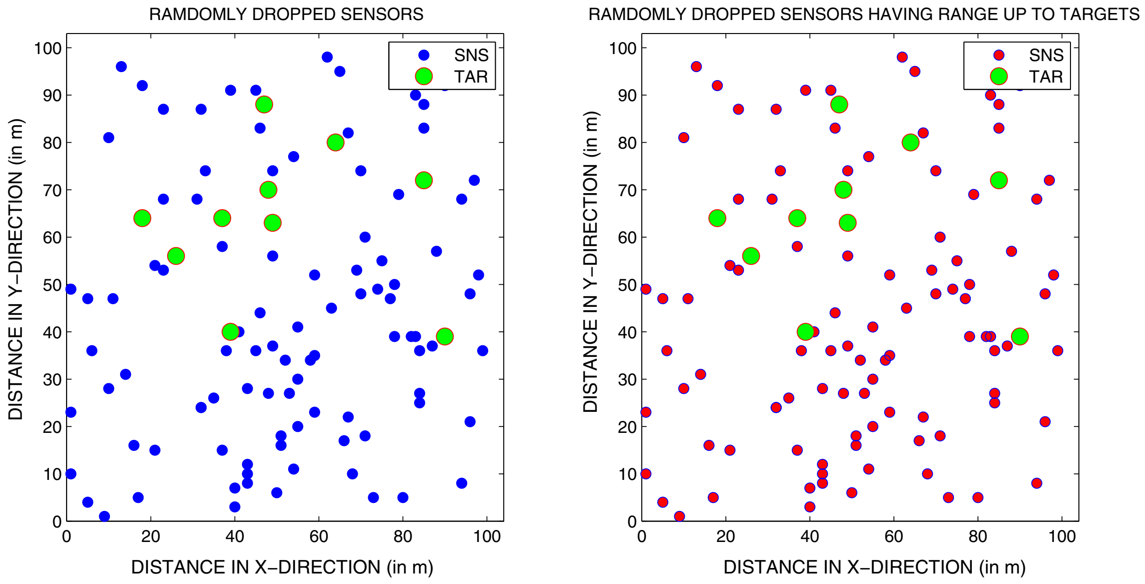


Fig. 9 WSN simulation development for Case 2

Table 2 Performance of different algorithms for Case 2 with population size 100

Trail no.	EVGM		EVCH		EVHM		EVHMRR	
	DC	RS	DC	RS	DC	RS	DC	RS
1	33	1	34	2	35	0	36	1
2	34	0	34	1	34	0	35	0
3	34	0	33	0	34	1	35	3
4	35	0	34	0	35	0	36	3
5	34	1	34	0	34	0	36	5

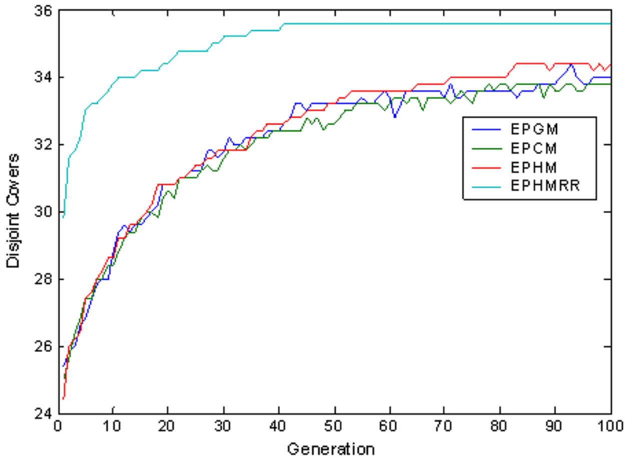


Fig. 10 Mean convergence characteristics of different algorithm for Case 2 with population size 100

Table 3 Performance of different algorithms for Case 2 with population size 200

Trail no.	EVGM		EVCH		EVHM		EVHMRR	
	DC	RS	DC	RS	DC	RS	DC	RS
1	34	0	34	0	35	1	36	1
2	34	0	34	0	36	0	36	0
3	34	0	36	0	35	1	36	0
4	34	0	35	1	36	0	36	4
5	34	0	34	1	35	0	36	2

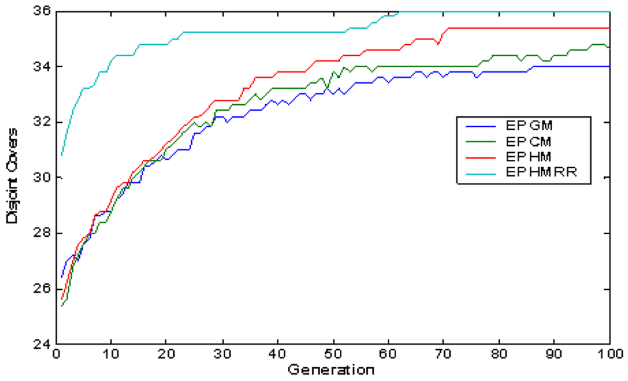


Fig. 11 Mean convergence characteristics of different algorithm for Case 2 with population size 200

upper bound of 36 with fewer sensors; however, as in the first case, EPHMRR outperformed the other algorithms, which was due to its hybrid mutation and redundancy removal capabilities.

5.1 Algorithmic Approach

Step 1: A population of N trial solutions was initialized. Each solution was taken as a pair of real-valued vectors $(\bar{x_i}, \bar{\sigma_i})$, $\forall i \in \{1, 2, 3, \ldots, N\}$ with their dimensions corresponding to the number of variables. The initial components of each $\bar{x_i}, \forall i \in \{1, 2, 3, \ldots, N\}$ were selected in accordance with a uniform distribution ranging over a presumed solution space. The values of $\bar{\sigma_i}, \forall i \in \{1, 2, 3, \ldots, N\}$, the strategy parameters, were initialized with some random values.
Step 2: The fitness score for each solution $\bar{x_i}$ was evaluated in light of an objective function $\phi(\bar{x_i})$.
Step 3: Two offspring were generated from each parent, and they are given by Eqs. (5) and (6). Self adaptiveness is given by Eq. (7).

$$\bar{x}'_i(j) = \bar{x}_i(j) + \bar{\sigma}_i(j) C_j \tag{5}$$

$$\bar{x}''_i(j) = \bar{x}_i(j) + N_j(0, 1) \tag{6}$$

$$\bar{\sigma}'_i(j) = \bar{\sigma}_i.exp(\tau' N(0, 1) + \tau N_j(0, 1)) \tag{7}$$

$\forall j \in \{0, 1, 2\ldots, r\}$
where, $\bar{x}'_i(j), \bar{x}_i(j), \bar{x}''_i(j), \bar{\sigma}'_i(j)$ and $\bar{\sigma}_i(j)$ denote the jth component of vectors $\bar{x}'_i, \bar{x}_i, \bar{x}''_i, \bar{\sigma}'_i$ and $\bar{\sigma}_i$ respectively. C_j is the Cauchy random variable with scale parameter t = 1. N(0,1) denotes a standard Gaussian random variable. $N_j(0, 1)$ indicates that the random variable is sampled for each new value of the counter j. The scaling factors τ and τ' are robust exogenous parameters which have been set to $(\sqrt{2\sqrt{n}})^{-1}$ and $(\sqrt{2n})^{-1}$ respectively.
Step 4: With developed offspring, the chromosome correction process is applied if there is a requirement to keep the sensors within the specified value.
Step 5: The fitness score for each offspring $\phi(\bar{x_i})$ is determined.
Step 6: Comparisons of fitness were made between two developed offspring to determine the best one, and the process was repeated to create a population of offspring.
Step 7: To develop a total of 2N solutions for each solution, 10% of the population was randomly selected as opponents from among all parents and offspring with equal probability. In each comparison, if the conditioned solution had at least as good a performance as the randomly selected opponent, it received a win.
Step 8: The N best solutions out of 2N, based on the number of wins received, were selected to be the parents of the subsequent generation.
Step 9: The algorithm proceeded to step 3 if the available time for execution ran out or the acceptable solution was obtained.

5.2 Redundancy Removal Operation with Forward Scanning

Redundant sensors were used to replace malfunctioning sensors in the formed disjoint covers. The process of using redundant sensors based on necessity is called Forward scanning. The redundancy removal algorithm is given in Algorithm 1.

Algorithm 1 Redundancy Removal

```
$P_r = [\,]$;
while No. of sensed Target $\neq$ Total Available Target do {
    $X_n \leftarrow$ A new sensor node from solution
    $C_n \leftarrow$ Total defined targets with $X_n$ nodes
    $X_r = C_n \oplus P_r$;
    $d_1 = \sum |X_r - Cn|$;
    $d_2 = \sum |X_r - Pr|$;
    $s_1 = \sum C_n$;
    $s_2 = \sum P_r$;
    if $(d_1 == 0)\&(s_2 < s_1)$ then
        Selected Sensor Node $X_n$ is redundant;
        Remove from there and place at the end
    end if
    if $(d_2 == 0)$ then
        All previous sensor nodes in the cover are redundant;
        Remove redundant nodes from there and place at the end
    end if
    $P_r = X_r$ }
end while
```

Case 1: Sensing range = 20 m; targets = 15; deployed sensors number = 200; and upper bound of cover = 37

Case 2: Sensing range = 50 m; targets = 10; deployed sensors number = 100; and upper bound of cover = 36

6 Result Analysis and Inference

Our results show that the hybrid evolutionary programming approach is one of the best techniques for creating disjoint sensor covers to increase the lifetime of WSNs. Our approach of increasing the lifetimes of WSNs by using algorithms that employ hybrid evolutionary programming is shown in terms of disjoint covers (DC) and redundant sensors (RS) in Tables 1, 2 and 3 for different sets of input data. Comparative performance analyses of the four algorithms EVGM, EVCH, EVHM and EVHMRR are also shown in Figs. 7, 8, 10 and 11. For different sets of input data, it can be observed that solution convergence, the number of disjoint covers created and the number of redundant sensors that can replace similar malfunctioning sensors in the active sensor cover increase from the first algorithm, EVGM, to the last algorithm, EVHMRR. Thus, we have shown in detail the performance comparisons of the proposed algorithms, and to the best of our knowledge, this study is the first to use algorithms with hybrid evolutionary programming to enhance the lifetime of WSNs (Fig. 9).

7 Conclusion

In this paper, a method that uses a hybrid evolutionary approach to develop a maximum number of possible disjoint covers is proposed to enhance the lifetimes of wireless sensor networks. The hybrid approach combines Gaussian and Cauchy distributions to develop better offspring solutions. The Cauchy distribution is useful in the early stages of solution development because it delivers a large change, while the Gaussian distribution, which delivers a small change, is important for the later stage of solution development. Here, a mechanism is developed to remove the redundancy present in disjoint covers in the process of solution development, and it achieves the upper bound of the disjoint cover in a smooth and optimal manner. As a future direction, it will be interesting to investigate other optimization techniques, with an emphasis on those that use a hybrid distribution approach.

References

1. Abrams, Z., Goel, A., Plotkin, S.: Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In: Proceedings of the 3rd international symposium on Information processing in sensor networks, pp. 424–432. ACM (2004)
2. AlShawi, I.S., Yan, L., Pan, W., Luo, B.: Lifetime enhancement in wireless sensor networks using fuzzy approach and a-star algorithm. Sensors Journal, IEEE **12**(10), 3010–3018 (2012)
3. Anitha, R., Kamalakkannan, P.: Enhanced cluster based routing protocol for mobile nodes in wireless sensor network. In: Pattern Recognition, Informatics and Mobile Engineering (PRIME), 2013 International Conference on, pp. 187–193. IEEE (2013)
4. Bahi, J., Haddad, M., Hakem, M., Kheddouci, H.: Efficient distributed lifetime optimization algorithm for sensor networks. Ad Hoc Networks **16**, 1–12 (2014)
5. Berger-Wolf, T.Y., Hart, W.E., Saia, J.: Discrete sensor placement problems in distribution networks. Mathematical and Computer Modelling **42**(13), 1385–1396 (2005)
6. Chen, Z., Li, S., Yue, W.: Sofm neural network based hierarchical topology control for wireless sensor networks. Journal of Sensors **2014** (2014)
7. Cheng, Y., Yang, L.: A novel energy-efficient reception method based on random network coding in cooperative wireless sensor networks. International Journal of Distributed Sensor Networks **2015**, 1 (2015)
8. Dietrich, I., Dressler, F.: On the lifetime of wireless sensor networks. ACM Transactions on Sensor Networks (TOSN) **5**(1), 5 (2009)
9. Fan, G., Jin, S.: Coverage problem in wireless sensor network: A survey. Journal of networks **5**(9), 1033–1040 (2010)
10. Guerriero, F., Violi, A., Natalizio, E., Loscri, V., Costanzo, C.: Modelling and solving optimal placement problems in wireless sensor networks. Applied Mathematical Modelling **35**(1), 230–241 (2011)
11. Halder, S., Ghosal, A., Saha, A., DasBit, S.: Energy-balancing and lifetime enhancement of wireless sensor network with archimedes spiral. In: Ubiquitous Intelligence and Computing, pp. 420–434. Springer (2011)
12. Kumar, P., Chaturvedi, A.: Life time enhancement of wireless sensor network using fuzzy c-means clustering algorithm. In: Electronics and Communication Systems (ICECS), 2014 International Conference on, pp. 1–5. IEEE (2014)
13. Li, H., Miao, H., Liu, L., Li, L., Zhang, H.: Energy conservation in wireless sensor networks and connectivity of graphs. Theoretical Computer Science **393**(1), 81–89 (2008)
14. Meguerdichian, S., Koushanfar, F., Potkonjak, M., Srivastava, M.B.: Coverage problems in wireless ad-hoc sensor networks. In: INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, vol. 3, pp. 1380–1387 vol.3 (2001). doi:10.1109/INFCOM.2001.916633
15. Mehra, M., Dabas, P.: Energy efficient secure routing protocol (eesrp) in wireless sensor network. International Journal for Innovative Research in Science and Technology **2**(3), 29–33 (2015)
16. Padmavathy, T., Chitra, M.: Extending the network lifetime of wireless sensor networks using residual energy extraction-hybrid scheduling algorithm. International Journal of Communications, Network and System Sciences **3**(1), 98 (2010)
17. Pattani, K.M., Chauhan, P.J.: Energy saving in wireless sensor network with spin protocol (2015)
18. Rout, R.R., Ghosh, S.K.: Enhancement of lifetime using duty cycle and network coding in wireless sensor networks. Wireless Communications, IEEE Transactions on **12**(2), 656–667 (2013)
19. Sharawi, M., Emary, E., Saroit, I.A., El-Mahdy, H.: Bat swarm algorithm for wireless sensor networks lifetime optimization (2014)
20. Shu, H., Liang, Q., Gao, J.: Distributed sensor networks deployment using fuzzy logic systems. International Journal of wireless information networks **14**(3), 163–173 (2007)
21. Suh, C., Mir, Z.H., Ko, Y.B.: Design and implementation of enhanced ieee 802.15. 4 for supporting multimedia service in wireless sensor networks. Computer Networks **52**(13), 2568–2581 (2008)
22. Wang, C.F., Shih, J.D., Pan, B.H., Wu, T.Y.: A network lifetime enhancement method for sink relocation and its analysis in wireless sensor networks. Sensors Journal, IEEE **14**(6), 1932–1943 (2014)
23. Wang, F., Liu, J.: Networked wireless sensor data collection: issues, challenges, and approaches. IEEE Communications Surveys & Tutorials **13**(4), 673–687 (2011)
24. Wang, H., Agoulmine, N., Ma, M., Jin, Y.: Network lifetime optimization in wireless sensor networks. Selected Areas in Communications, IEEE Journal on **28**(7), 1127–1137 (2010)
25. Zorbas, D., Glynos, D., Kotzanikolaou, P., Douligeris, C.: Solving coverage problems in wireless sensor networks using cover sets. Ad Hoc Networks **8**(4), 400–415 (2010)

P. Nagarathna obtained her B.E. in Computer Science and Engineering from Visvesvaraya Technological University (VTU), Belgaum and M. E. in Computer Science and Engineering from Bharath University. Currently she is pursuing Ph.D. at the School of Computing Science and Engineering, VIT University, Vellore, India. Her interest includes Wireless Sensor Networks, Computer Networks, Data Structures and Objected Oriented Programming.

R. Manjula received her B.E in Computer Science and Engineering from University Visvesvaraya College of Engineering (UVCE), Bangalore, Karnataka State, India in 1992 and M.E in Software Engineering from Anna University, Tamil Nadu, India in 2001 and Ph.D. in Computer Science and Engineering from VIT University, Vellore, India. Presently she is working as Associate Professor at the School of Computing Science and Engineering at VIT University, Vellore, India. Her area of specialization includes Software Process modeling, Software Testing and Metrics, Service Oriented Architecture, Data Mining and Social Network Analysis.