

Nature-Inspired Algorithms for k -Coverage and m -Connectivity Problems in Wireless Sensor Networks



Subash Harizan and Pratyay Kuila

Abstract Efficient deployment of the sensor nodes takes an important role in proper coverage and connectivity of the wireless sensor networks (WSNs). The key issues that need to be taken care during deployment are the number of deployed sensors, coverage of the target/region, and connectivity among the sensor nodes. As the sensor nodes are prone to various kinds of failure, it is essential for k -coverage of the targets and m -connectivity among the sensor nodes. Here, k -coverage of the targets indicates that all the targets are covered by at least k number of sensor nodes so that failure of $k - 1$ sensor nodes can also ensure coverage of the targets. Similarly, m -connectivity of the sensor nodes indicates all the sensor nodes are connected with other $m - 1$ sensor nodes. Note that the k -coverage and m -connectivity problem for WSNs is nondeterministic polynomial (NP)-hard in nature. In this chapter, nature-inspired algorithms are studied and designed to solve the problem. Particle swarm optimization (PSO), differential evolution (DE), genetic algorithms (GA), and gravitational search algorithm (GSA) are studied and designed for the problem. The chromosome, vector, particle, and agent are efficiently represented. An efficient derivation of fitness functions is provided with the conflicting objectives. An extensive simulation is also conducted.

Keywords Nature-inspired algorithms · k -coverage · m -connectivity · Wireless sensor networks

S. Harizan · P. Kuila (✉)
Department of Computer Science & Engineering, National Institute of Technology Sikkim,
South Sikkim 737139, India
e-mail: pratyay_kuila@yahoo.com

S. Harizan
e-mail: subashharizan@gmail.com

© Springer Nature Singapore Pte Ltd. 2020
S. K. Das et al. (eds.), *Design Frameworks for Wireless Networks*,
Lecture Notes in Networks and Systems 82,
https://doi.org/10.1007/978-981-13-9574-1_12

1 Introduction

Wireless sensor networks (WSNs) draw a great attention of the researchers for their various potential applications. A sensor node performs multiple functions as sensing, communication, and data processing. Sensor nodes can collaborate among them to monitor or sense the area of interest (AoI), collect the sensory data, process the data, and transmit to the base station (BS) directly or through multi-hop [1–4]. As the sensor nodes are equipped with limited energy source, it is very essential and challenging to conserve the limited energy [5–9]. Moreover, sensor nodes can sense and communicate within the limited sensing and communication range. A region/target is said to be covered if it falls within the sensing range. Similarly, two sensor nodes can communicate with each other if they are within the communication range of each other. Therefore, it is also very essential to provide an efficient coverage and connectivity [10–16] to monitor the region/target and to transmit the sensed data to the base station.

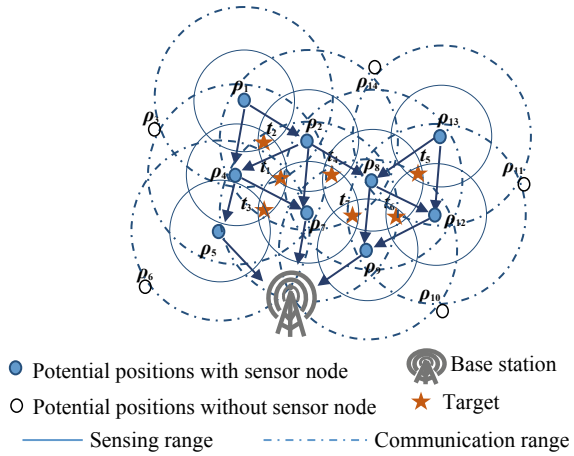
Deployment of sensor nodes has vital role for ensuring an efficient coverage and connectivity in the network. There are mainly two types of deployment scheme in WSN: preplanned [14] and ad hoc [14] deployment. In preplanned scheme, sensor nodes are deployed in planned manner in an accessible area and thereby the network has better management with saving in cost and energy. In ad hoc scheme, sensor nodes are deployed randomly in the harsh environment where human interference is not possible. Ad hoc deployment does not guarantee the coverage and connectivity. Thus, this scheme requires large number of sensor nodes for proposed function of the network.

Sensor nodes are prone to failure due to hardware failure, energy depletion, natural calamities, etc. As a result, network becomes functionless. Therefore, for better performance of the network k -coverage and m -connectivity are desirable. However, cost for the k -coverage and m -connectivity should also be optimized by deploying the minimum number of sensor nodes. Therefore, deployment of minimum number of sensor nodes for desire coverage and connectivity is an NP-hard [14, 16–18] problem. In limited time, it is not feasible or computationally very expensive to find the optimal solutions for the NP-hard problem.

In this chapter, we studied the target coverage with connectivity problem as in [14]. As an example, a network scenario with 7 target points ($t_1, t_2, t_3, \dots, t_7$) and 14 potential positions ($\rho_1, \rho_2, \rho_3, \dots, \rho_{14}$) is shown in Fig. 1. Here, among the 14 potential positions, 9 positions are selected to place the sensor nodes. We can also observe that all the target points are covered by at least three sensor nodes and each sensor node is connected with at least other two neighbor sensor nodes. Therefore, we can say that scenario of the network as shown in Fig. 1 in chapter “Ambient Intelligence for Patient-Centric Healthcare Delivery: Technologies, Framework and Applications” is three coverage of the target and two connectivity of the sensor nodes, i.e., 3-coverage and 2-connectivity.

Nowadays, nature-inspired algorithms (NIAs) are drawing a great attention of the researchers as these algorithms are found to be very efficient to find the optimal

Fig. 1 A simple network scenario with 3-coverage and 2-connectivity [14]



solutions for the real-life complex problems. NIAs are classified into evolutionary algorithms (EAs) and swarm intelligence algorithms (SIAs). These algorithms are extensively used to solve many optimization problems of WSNs [19–29].

1.1 Author’s Contribution

In this chapter, we have studied the k -covered and m -connectivity problem as in [14]. Various nature-inspired approaches like genetic algorithm (GA), particle swarm optimization (PSO), differential evolution (DE), and gravitational search algorithm (GSA) are also studied and employed for the above mentioned problem. Our contributions are summarized as follows:

- A linear programming (LP) is formulated for the aforesaid problem.
- Efficient representation of chromosome, particle, vector, and agent for GA, PSO, DE, and GSA, respectively. Moreover, they are generated such a way that validity of them cannot be disturbed after the operations of EAs (e.g., crossover, mutation, velocity and position update, etc.).
- Derivation of efficient fitness function is given. Here, three conflicting objectives are considered.
- An extensive simulation is conducted and comparisons are shown for the algorithms.

1.2 Organization of the Chapter

Rest of the chapter is organized as follows. Section 2 provides a brief overview of nature-inspired algorithms. Network model, problem formulation, terminologies, and derivation of fitness function are defined in Sect. 3. GA-, PSO-, DE-, and GSA-based approaches are discussed in Sects. 4–7, respectively. Experimental results are shown in Sect. 8 and the chapter is concluded in Sect. 9.

2 Nature-Inspired Algorithms

Nature-inspired algorithms (NIAs) are inspired by the processes, perceived from nature to solve the various optimization problems. These algorithms have drawn enormous attention of the researchers to solve various optimization problems in the field of engineering, biomedical, finance, etc. NIAs find the optimal solution for real-life problem which is classified as NP-hard. It is a population-based algorithm which is classified into evolutionary algorithms (EAs) and swarm-intelligence-based algorithms (SIAs). EAs are motivated by the theory of Charles Darwin called survival of the fittest. The individual in the population survives and reproduces offspring only if they can fit themselves in the given environment. Whereas SIAs are inspired by the collective behavior of swarms like bird flocking, fish schooling, ant colony, bee colony, etc. An overview of some popular nature-inspired approaches is as follows.

2.1 Genetic Algorithm

Genetic algorithm (GA) [30–34] is a population-based meta-heuristic optimization algorithm. A population of size N_p is created by randomly generating the chromosomes. Each chromosome represents the complete solution to the problem. Chromosomes in the population are updated by applying the genetic operations (selection, crossover, and mutation) to explore the search space to find the near-optimal solution. In the selection phase, a set of chromosomes are selected from the population. Selection operation is followed by the crossover operation. In crossover operation, two-parent chromosomes exchange their information to produce two child offspring chromosomes. Finally, a gene value is randomly selected and mutated. Mutation is also applied with some modification according to the application characteristic. This process repeats iteratively till the desired or satisfactory chromosome is obtained or maximum number of iterations.

2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) [1, 35, 36] is a stochastic optimization method that is motivated by the social behavior of bird flocking or fish schooling. It can be observed that birds, fishes, etc. always travel in a group without colliding with each other. To avoid the collision, each member uses the group information and adjusts its position and velocity to follow the group. This approach reduces the efforts of each individual to find the food, shelter, etc. Initially, a swarm (say size N_p) of particles is created randomly where an individual particle represents a complete solution to a multidimensional optimization problem. All particles in the population have equal dimension (say D) and also initiated with random velocities in search space. In the d th dimension of the hyperspace, a particle P_i , $1 \leq i \leq N_p$ with position $\beta_{i,d}$, $1 \leq d \leq D$ can be represented as follows:

$$P_i = [\beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \dots, \beta_{i,d}] \quad (1)$$

The quality of the particles is evaluated by the derived fitness function. In order to search the near-optimal solution, the velocities and positions of the particles are updated in each iteration. The velocity of the particles is updated by two best particles, i.e., personal best ($Pbest_i$) and global best ($Gbest$). $Pbest_i$ be the best particle that has been observed so far for P_i . $Gbest$ be the best particle among the swarm. The velocity and position of the particles are updated as follows:

$$v_{i,d}(t) = w \times v_{i,d}(t-1) + c_1 \times r_1 \times (Pbest_{i,d} - \beta_{i,d}(t-1)) + c_2 \times r_2 \times (Gbest_{i,d} - \beta_{i,d}(t-1)) \quad (2)$$

$$\beta_{i,d}(t) = \beta_{i,d}(t-1) + v_{i,d}(t) \quad (3)$$

where w represents the inertial weight, c_1 and c_2 are positive constant values called acceleration coefficient, and r_1 and r_2 are two independently generated random number in the range [0, 1]. The update process will continue till the acceptable solution is achieved or the maximum number of iterations are reached.

2.3 Differential Evolution

Differential evolution (DE) [37–39] is one of the most powerful stochastic real parameter based evolutionary algorithms. DE is widely used to solve many optimization problems. Initially, a population (size say N_p) is created by randomly generating the set of vectors of dimension, say D . Each individual vector represents the complete solution to the problem. The quality of the vectors is evaluated by the derived fitness function. After the initialization of the population, the quality of the vectors is

enhanced by the mutation, crossover, and selection operation. In each iteration, the vectors are updated. The final solution is obtained by evaluating the solutions till the maximum iteration is reached.

DE with different variants can be represented in general form as $DE/\beta/\gamma/\delta$. DE denotes the differential evolution, β specifies the vector to be muted (which can be selected as randomly or best vector from population), γ specifies the number of vectors that are considered for mutation operation of β , and δ denotes crossover scheme (binomial or exponential). In the mutation operation, each candidate vector called target vector and other three vectors are randomly chosen. Thereafter, a new vector called donor vector is generated using mutation. Mutation operation is followed by crossover operation. In crossover operation, target vector and donor vector exchange their information to generate a child vector called trail vector. Finally, fitness function is used in selection operation to select the best vector among the trial vector and target vector for the next generation. Algorithm iterates continuously to generate the new vectors till the termination criteria are obtained or acceptable vector for the problem is achieved.

2.4 Gravitational Search Algorithm

Gravitational search algorithm (GSA) [40] is a population-based stochastic optimization technique. This algorithm is inspired from the law of gravitation and law of motion. In GSA, each solution is represented by agent that represents the complete solution for a multidimensional optimization problem. The agents are initialized with position, velocity, acceleration, and mass. According to the law of gravitation, objects in the universe attract each other by a gravitational force. During this movement, object having lower mass moves toward the object with higher mass. The objects with lower mass have higher acceleration whereas objects with higher mass have slower acceleration. The agents with lighter mass will get attracted toward the agent with heaviest mass. Performance of the agent is dignified by its mass. An agent with heavier mass is considered as the optimal solution to the problem. An i th agent from the population of size N_p is denoted as follows:

$$\lambda_i = \{\lambda_i^1, \lambda_i^2, \lambda_i^3, \dots, \lambda_i^d\}, \quad \forall i, 1 \leq i \leq N_p \quad (4)$$

where λ_i^d represents the position value of i th agent in the d th dimension. At iteration t , the gravitational force acting on agent λ_i from agent λ_j in the d th dimension is defined as follows:

$$F_{ij}^d = G(t) \times \frac{M_{pi}(t) \times M_{aj}(t)}{R_{ij}(t) + \varepsilon} \times \{\lambda_i^d(t) - \lambda_j^d(t)\} \quad (5)$$

where $G(t)$ is the gravitational constant at iteration t , ε is the small constant, passive gravitational mass of agent λ_i and active gravitational mass of agent λ_j are represented

by $M_{pi}(t)$ and $M_{aj}(t)$, respectively, and $R_{ij}(t)$ denotes the Euclidean distance between two agents λ_i and λ_j at iteration t . The value of gravitational constant $G(t)$ with initial value G_0 is calculated as follows:

$$G(t) = G_0 \times (-\beta t) / e^{tmax} \quad (6)$$

where $tmax$ denotes the maximum number of iteration and value of control parameter is represented by β . The total force applied by all agents on the agent λ_i in the d th is calculated by the following formula:

$$F_i^d = \sum_{j=1, j \neq i}^{Np} rand_j \times F_{ij}^d, \text{ where } rand_j \in [0, 1] \quad (7)$$

The relation among mass (M), acceleration (a), and force is given by the law of motion. Therefore, the acceleration of λ_i in d th dimension is given as follows:

$$a_i^d = F_i^d(t) / M_{ii}(t) \quad (8)$$

where M_{ii} denotes the inertial mass of an agent λ_i . The velocity and position of λ_i are updated by using the following equations:

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \quad (9)$$

$$\lambda_i^d(t+1) = \lambda_i^d(t) + v_i^d(t+1) \quad (10)$$

where $rand_i \in [0, 1]$. The gravitational and inertial masses of an agent are assumed to be equal in GSA, i.e., $M_{ai} = M_{pi} = M_{ii} = M_i$.

In a population, N_p agents are evaluated by the fitness function. In iteration t , among all agents, $best(t)$ and $worst(t)$ are the best and worst agents for the maximization problem denoted as

$$best(t) = \max_{j \in [1, \dots, Np]} fit_j(t) \quad (11)$$

$$worst(t) = \min_{j \in [1, \dots, Np]} fit_j(t) \quad (12)$$

The mass of every agent is calculated by the fitness value obtained from Eqs. (11) and (12). The mass M_i of an agent λ_i is determined as follows:

$$m_i = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad (13)$$

$$M_i = \frac{m_i(t)}{\sum_{j=1}^{Np} m_i(t)} \quad (14)$$

The exploitation must fade in and exploration must fade out to avoid from trapping in local optimum as the laps of iterations continue. Therefore, only K agents with best fitness value, i.e., $Kbest$ will apply force to others. $Kbest$ value linearly decreases to one as it is a function of time. Thus, Eq. (7) could be revised as follows:

$$F_i^d = \sum_{j \in Kbest, j \neq i} rand_j \times F_{ij}^d, \text{ where } rand_j \in [0, 1] \quad (15)$$

However, the subsequent changes in the position and velocity occur with the process of repeatedly updating the acceleration of the agent. This process of updating continues till the optimal solution is obtained or maximum iteration is reached.

3 Network Model and Problem Formulation

3.1 Network Model

Based on our problem, we have assumed a 2D WSN with few target points, a base station, and few predefined potential positions. Target points need to be monitored by placing the sensor nodes on the given potential positions. Assumed WSN is supposed to have the following properties [41–46]:

- All the target points and deployed sensor nodes are stationary.
- Target is said to be covered if it falls within the sensing range of sensor nodes.
- Two nodes are connected if they are within communication range of each other.
- A sensor node can sense more than one target point.
- All the sensor nodes have same sensing and communication range.

3.2 Terminologies

Before formulation of problem, first, we define some terminologies used in this chapter.

- $P = \{\rho_1, \rho_1, \rho_1, \dots, \rho_N\}$ denotes the N number of predefined potential positions.
- $S = \{s_1, s_2, s_3, \dots, s_Z\}$ denotes the set of Z number of deployed sensor nodes on selected potential positions.
- $T = \{t_1, t_2, t_3, \dots, t_M\}$ denotes the M number of target points.
- R_{sen} and R_{com} denote the sensing and communication range of the sensor nodes, respectively.
- $D(s_i, t_j)$ represents the Euclidean distance between s_i and t_j .
- $SC_{cov}(t_i)$ denotes the set of sensor nodes that cover t_i , i.e.,

$$SC_{cov}(t_i) = \{s_j | D(s_i, t_j) \leq R_{sen}, \forall j, 1 \leq j \leq Z\} \quad (16)$$

- $TC_{cov}(s_i)$ denotes the set of target points which are covered by s_i , i.e.,

$$TC_{cov}(s_i) = \{t_j | D(s_i, t_j) \leq R_{sen}, \forall j, 1 \leq j \leq M\} \quad (17)$$

- $C_{con}(s_i)$ denotes the set of sensor nodes that are within the communication range of s_i toward BS, i.e.,

$$C_{con}(s_i) = \{s_j | D(s_i, s_j) \leq R_{com}, \& D(s_i, s_j) \geq D(s_j, BS), \forall j, 1 \leq j \leq Z\} \quad (18)$$

- $COV_{cost}(t_i)$ denotes the coverage cost of t_i .
- $CON_{cost}(s_i)$ denotes the connection cost of s_i .
- k and m are the desired coverage and connectivity (k and m are some predefined value).
- BS denotes base station.

3.3 Problem Definition

Given a WSN with N predefined potential positions and M target points, we have to place the sensor nodes on selected potential positions, considering the following objectives:

- Selection of minimum number of potential positions for placement of sensor nodes.
- Placed sensor nodes must ensure the k -coverage of the targets.
- Placed sensor nodes must be m -connected among themselves.

Before formulation of linear programming (LP) of the given problem, we define the following Boolean variables:

$$\lambda_i = \begin{cases} 1, & \text{If a sensor node is placed at } \rho_i \\ 0, & \text{Otherwise} \end{cases} \quad (19)$$

$$\beta_{ij} = \begin{cases} 1, & \text{If } s_j \text{ provides coverage to } t_i \\ 0, & \text{Otherwise} \end{cases} \quad (20)$$

$$\partial_{ij} = \begin{cases} 1, & \text{If } s_i \text{ is within } R_{com} \text{ of } s_j \\ 0, & \text{Otherwise} \end{cases} \quad (21)$$

$$\delta_i = \begin{cases} 1, & \text{If } \rho_i \text{ is within } R_{com} \text{ of } BS \\ 0, & \text{Otherwise} \end{cases} \quad (22)$$

Now, the LP of the given problem is defined as follows:

$$\text{Minimize } \sum_{i=1}^N \lambda_i \quad (23)$$

Subject to:

$$\sum_{j=1}^Z \beta_{ij} \times \lambda_j \geq k, \text{ where } i = 1 \text{ to } M \quad (24)$$

$$\left(\sum_{j=1}^Z \partial_{ij} \times \lambda_j + \delta_i \right) \geq m, \text{ where } i = 1 \text{ to } Z \quad (25)$$

$$(\lambda_j, \beta_{ij}, \partial_{ij}, \delta_i) \in \{0, 1\} \quad (26)$$

Constraint 24 ensures the k -coverage of each target point and m -connectivity among the deployed sensor nodes is ensured by the constraint 25. Restriction on the decision variables is given by constraint 26.

3.4 Derivation of Fitness Function

In this chapter, we have studied four nature-inspired algorithms. The algorithms are initialized with population which consists of member of solutions (i.e., chromosomes/vectors/particles/agents). The solutions are evaluated on the basis of fitness function. The following objectives are considered as in [14].

Objective 1 (*Deployment of minimum number of sensor nodes*): Let us assume that out of given N potential positions, Z numbers of potential positions are selected by particular solution to place the sensor nodes. The first objective is as follows:

$$\text{Objective 1: Minimize } O_1 = Z/N \quad (27)$$

Objective 2 (*Maximization of k -coverage*): The second objective is to maximize the k -coverage of all the target (M) points in the network which can be stated as follows:

$$\text{Objective 2: Maximize } O_2 = \frac{\sum_{i=1}^M COV_{cost}(t_i)}{(M \times k)} \quad (28)$$

where coverage cost ($COV_{cost}(t_i)$) of t_i is defined as follows:

$$COV_{cost}(t_i) = \begin{cases} k, & \text{If } |SC_{cov}(t_i)| \geq k \\ SC_{cov}(t_i), & \text{Otherwise} \end{cases} \quad (29)$$

Objective 3 (*Maximization of m -connectivity*): Deployed sensor nodes on the selected potential positions must be m -connected and toward BS. The third objective can be stated as follows:

$$\text{Objective 3: Maximize } O_3 = \frac{\sum_{i=1}^Z CON_{cost}(s_i)}{(Z \times m)} \quad (30)$$

where connection cost ($CON_{cost}(s_i)$) of s_i is given as

$$CON_{cost}(t_i) = \begin{cases} m, & \text{If } |C_{con}(s_i)| \geq m \\ C_{con}(s_i), & \text{Otherwise} \end{cases} \quad (31)$$

The above objectives are conflicting in nature. Weight sum approach (WSA) [47] is found to be very efficient to form a single fitness function taking multiple multi-objective functions. Thus, WSA scheme is used as follows:

$$F = w_1 \times (1 - O_1) + w_2 \times O_2 + w_3 \times O_3 \quad (32)$$

$$\begin{aligned} \text{i.e., } F = & w_1 \times (1 - (Z/N)) + w_2 \times \frac{\sum_{i=1}^M COV_{cost}(t_i)}{(M \times k)} \\ & + w_3 \times \frac{\sum_{i=1}^Z CON_{cost}(s_i)}{(Z \times m)} \end{aligned} \quad (33)$$

where $w_1 + w_2 + w_3 = 1$ and $0 \leq w_i \leq 1, \forall i, i = 1$ to 3. The final objective is to maximize the fitness value, i.e.,

$$\text{Objective} = \text{Maximize } F \quad (34)$$

Based on this fitness value as defined in Eq. 33, the solutions are evaluated. Solution with higher fitness value is considered as the better solution.

4 GA-Based Approach

4.1 Chromosome Encoding

The chromosomes are encoded as in [14]. Here, the chromosome as a string of zeros and ones is taken. The number of potential positions in the network is taken as the length of the chromosomes. If the value of i th gene is 1 then it implies that the ρ_i is selected for the placement of sensor nodes. Otherwise, no sensor nodes are placed at ρ_i .

Fig. 2 Chromosome representation

ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	ρ_9
1	0	1	0	1	0	1	0	1

Illustration: Let us consider a WSN with nine potential positions. Therefore, length of the chromosome is nine as in Fig. 2. It can be observed that gene value at position 5 is 1 which represents that the sensor node is placed at position ρ_5 . Similarly, sensor nodes are also placed at ρ_1 , ρ_3 , ρ_7 , and ρ_9 . Whereas no sensor nodes are placed at ρ_2 , ρ_4 , ρ_6 , and ρ_8 as the gene value at positions 2, 4, 6, and 8 is 0.

4.2 Initialization of Population

The initial population is randomly generated set of the chromosomes.

4.3 Fitness Function

Chromosomes are evaluated on the basis of the fitness function as defined in Sect. 2.4, Eq. 31.

4.4 Selection, Crossover, and Mutation Operation

In selection phase, two valid chromosomes called parents are selected from the population which can undergo crossover operation to produce new offspring. There are many selection methods like Roulette-wheel selection, rank selection, tournament selection, etc. Here, Roulette-wheel selection method is used.

Crossover operation is performed on two selected chromosomes. This operation is regulated by the crossover probability. Evolution of the search speed is reformed by varying this crossover probability. There are different types of crossover like one-point crossover, two-point crossover, uniform crossover, etc. Crossover points are chosen randomly in the chromosome beyond that the parent chromosomes exchange their information to produce new child chromosomes. Here, two-point crossover operation is used.

The crossover operation is followed by the mutation operation. In mutation operation, a randomly selected gene value within a chromosome is altered to produce a new species with arbitrary locus in the fitness landscape. Like crossover, performance of mutation operation is also regulated by the mutation probability. Mutation probability is usually lower than crossover probability. Mutation produces a chromosome which cannot converge in the local optimum.

5 PSO-Based Approach

5.1 Particle Representation

It is very essential to represent the particles in such a way that an individual particle must represent the complete solution to the problem. The dimensions of the particle are taken same as the number of potential positions (i.e., N) in the network. A swarm is a set of randomly generated N_p number of particles. The i th particle of the population may be represented by a vector P_i as follows:

$$P_i = [\beta_{i,1}, \beta_{i,2}, \beta_{i,3}, \dots, \beta_{i,N}]$$

where each component is given as $\beta_{i,d}$, $0 \leq i \leq Np$, $1 \leq d \leq N$. Here, $\beta_{i,j}$ represents the j th component of the i th particle. Each component of the particles is initialized by randomly generated uniformly distributed number $rand(0, 1)$, $0 < rand(0, 1) \leq 1$. Here, if the component value (say j th component) of the particle is greater than the defined threshold value (Th) then the corresponding potential position (j th) is being placed with a sensor node. Otherwise, sensor node is not placed at the potential position.

Illustration: Let us consider a WSN with nine potential positions. Therefore, dimension of the particle is same as the number of potential positions, i.e., nine. Figure 3 shows the particle representation of the corresponding WSN. Now, random numbers are generated for each component of the particle. We can also define a threshold value Th (say 0.54) which is compared with each component value. If the generated i th component value is greater than the Th value then the corresponding i th position is selected to place the sensor node. From Fig. 3, it can be observed that the component value at positions 1, 2, 3, 5, 7, and 9 has value greater than the defined Th . Therefore, the potential positions $\rho_1, \rho_2, \rho_3, \rho_5, \rho_7$, and ρ_9 are chosen for placement of sensor nodes. However, no sensor nodes are placed at potential positions ρ_4, ρ_6 , and ρ_8 as the component value at that 4, 6, and 8 is less than Th , i.e., 0.54.

5.2 Fitness Function

Here, same fitness function is used as discussed in Sect. 3.4, Eq. 31.

ρ_1	ρ_2	ρ_3	ρ_4	ρ_5	ρ_6	ρ_7	ρ_8	ρ_9
0.67	0.79	0.89	0.33	0.59	0.21	0.88	0.41	0.58

Fig. 3 Particle representation

5.3 Velocity and Position Update

In each iteration, the velocity and position of the particles are updated using Eqs. 2 and 3, respectively. Similarly, $Pbest$ and $Gbest$ are also updated accordingly.

6 DE-Based Approach

6.1 Vector Representation

The vectors are encoded same as the particles in PSO.

6.2 Fitness Function

The fitness function is used same as in GA and PSO.

6.3 Mutation

For the crossover and mutation operation, the *DE/best/1/bin* scheme is used. In differential mutation operation (say at t th generation for i th vector), a donor vector ($v_{i,t}^{donor}$) is created for each member vector (target vector) ($v_{i,t}^{tar}$) in the population. The best vector ($v_{i,t}^{best}$) and other two distinct vectors ($v_{x,t}$ and $v_{y,t}$) are randomly chosen from the population such that $i \neq x \neq y \neq best$. After that, difference of distinct vectors is multiplied with scaling factor F and added to best vector to generate a donor vector. The mutation operation can be defined as follows:

$$v_{i,t}^{donor} = v_{i,t}^{best} + F(v_{x,t} - v_{y,t}) \quad (35)$$

6.4 Crossover

Crossover operation is performed to generate a new vector called trail vector $v_{i,t}^{trial}$. Here, binomial crossover operation is performed in between donor vector and target vector. We choose a predefined crossover rate (say CR) and operation for crossover can be defined as follows:

$$v_{i,t}^{trial} = \begin{cases} v_{i,t}^{donor}, & \text{if } (rand[0, 1] \leq CR) \\ v_{i,t}^{tar}, & \text{Otherwise} \end{cases} \quad (36)$$

Each component of the trail vector is generated as follows. For each component, we choose a random value between 0 and 1. Say for i th component of a trail vector a chosen random value is less than or equal to CR then i th component of trail vector is same as the i th component of the donor vector. Otherwise, i th component of trail vector is same as the i th component of target vector.

6.5 Selection

Survival among the trail vector and target vector which acts as a target vector $v_{i,t+1}^{tar}$ in the next generation ($t + 1$) is decided by the selection operation. Derived fitness function as defined in Eq. 31 is used to evaluate both the vectors. If the fitness of the trail vector is found to be better than target vector then it will replace the target vector for the next generation. Otherwise, target vector will be part of next generation. The selection operation is defined as follows:

$$v_{i,t+1}^{tar} = \begin{cases} v_{i,t}^{trial}, & \text{if } (f(v_{i,t}^{trial}) \geq f(v_{i,t}^{tar})) \\ v_{i,t}^{tar}, & \text{Otherwise} \end{cases} \quad (37)$$

where $f(v_{i,t}^{trial})$ and $f(v_{i,t}^{tar})$ represent the fitness of the trail vector and target vector for the generation t .

7 GSA-Based Approach

7.1 Agent Representation

Agents are represented in the same way as the particle.

7.2 Update Velocity, Mass, Position, and Force

The velocity, mass, position, and force of the agents are updated by Eqs. 9, 14, 10, and 15, respectively.

7.3 Fitness Function

Fitness function is used same as in GA, PSO, and DE.

8 Experimental Results

We have performed extensive simulation of the algorithms using MATLAB and C programming. We have considered the scenario for random deployment where potential positions are randomly taken in the network area of $300 \times 300 \text{ m}^2$. BS is located at the position (300, 150) as shown in Fig. 4. The target points are denoted by the black triangle, potential positions are denoted by red circle, and selected potential points are denoted by blue circle. The used simulation parameters for PSO, DE, GA, and GSA are given in Table 1. It should be noted that it is very hard to

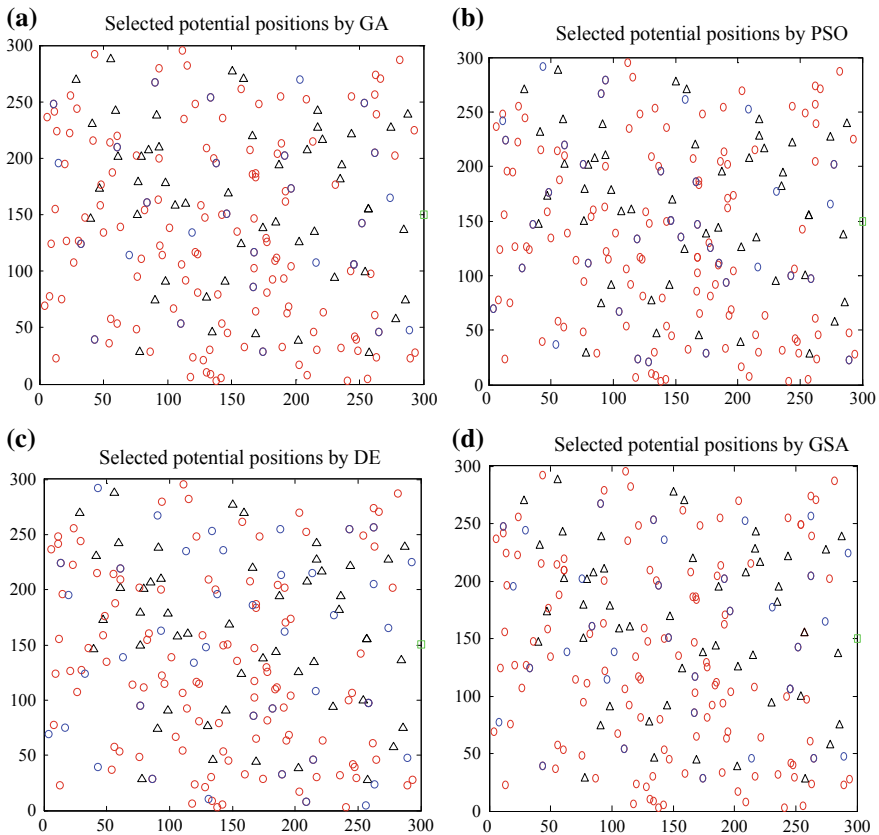


Fig. 4 Selection of potential positions by **a** GA, **b** PSO, **c** DE, and **d** GSA

Table 1 Simulation parameters

PSO		DE		GA	GSA	
c_1	1.4962	CR	0.7	Mutation rate 3%	G_0	3
c_2	1.4962	F	0.5		β	1
w	0.7968					
v_{max}	0.5					
v_{min}	-0.5					

precisely and accurately finalize the weight values. Therefore, we have tested for various combinations of the weight values and found a better result for $w_1 = 0.3$, $w_2 = 0.3$, and $w_3 = 0.4$. Therefore, we have taken the same.

We first execute the algorithms for $k = 1$ and $m = 1$ with randomly placed 50 target points and 150 potential positions. We have taken the sensing and communication range as 40 meters and 70 meters, respectively. The selected potential positions for the algorithms are shown in Fig. 4a–d. It can be seen that the PSO, DE, GA, and GSA select 36, 41, 27, and 33 potential positions, respectively. The objective 1 of the derived fitness function forces to select minimum number of sensor nodes. Other two fitness functions ensure coverage and connectivity.

We have also executed the simulation by varying the value of k from 1 to 3 and m from 1 to 3. The number of selected potential positions by the algorithms with varying the number of target points is shown in Fig. 5a–c. From Fig. 5a–c, it can also be observed that as the number of target points increases corresponding potential positions also increases for different values of k and m .

9 Conclusion

In this chapter, we have presented four nature-inspired algorithms, namely, GA-, PSO-, DE-, and GSA-based approach for the deployment of sensor nodes. The objectives for the deployment of sensor nodes are as selection of minimum number of potential positions for the placement of sensor nodes such that all the target points are k -covered along with m -connectivity among the placed sensor nodes. Linear programming is also formulated. Efficient representations of chromosome, particle, vector, and agents are illustrated for GA, PSO, DE, and GSA, respectively. To evaluate the solutions, derivation of an efficient fitness function is given by considering all the objectives. An extensive simulation is conducted for all the algorithms by varying the number of k and m , and target points. As all the algorithms are executed with the same derived fitness function, it is very hard to conclude and compare the performance among them. While, for a particular scenario, GA is proving better performance, for some other scenario, GSA may provide better.

In large industries, the scheme can be used to monitor some critical points like gas leakage, fire zone, etc. As the problem is multi-objective in nature, some suitable

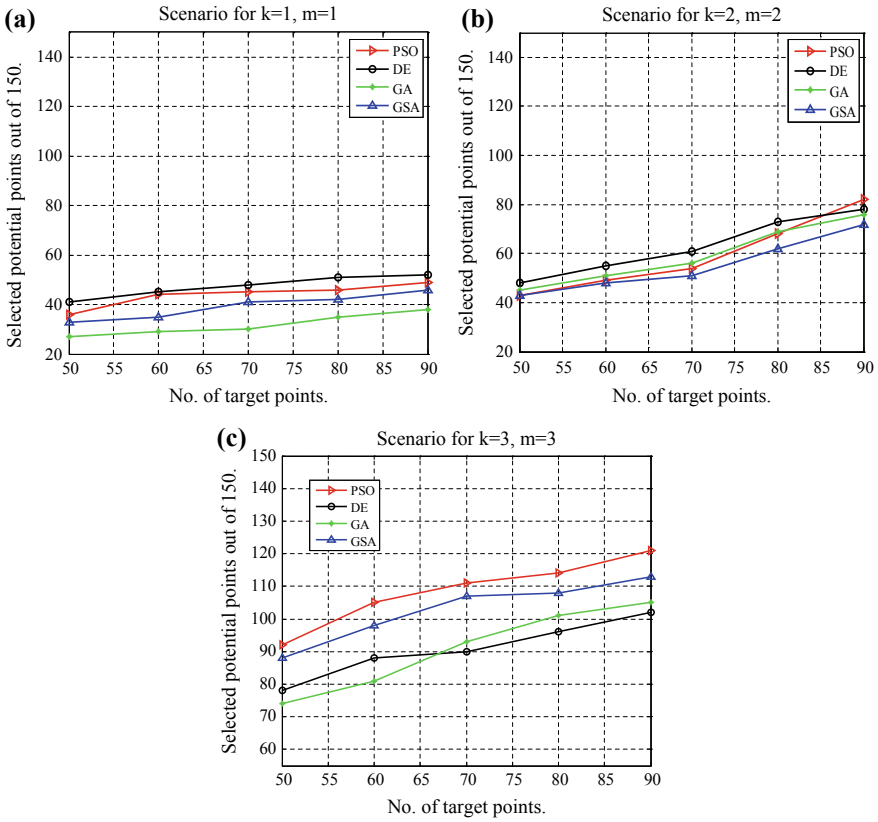


Fig. 5 Selection of potential positions by varying the number of target points for **a** $k = 1, m = 1$, **b** $k = 2, m = 2$, and **c** $k = 3, m = 3$

multi-objective evolutionary algorithm (MOEA) may be employed for the same. Mobility of the sensor nodes is not considered. The works may be extended by considering the mobility of the sensor nodes. Moreover, energy consumption of the sensor nodes is also not considered.

References

1. Kuila, P., & Jana, P. K. (2014). Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Engineering Applications of Artificial Intelligence*, 33, 127–140.
2. El-Fouly, F. H., Ramadan, R. A., Mahmoud, M. I., & Dessouky, M. I. (2018). REBTAM: Reliable energy balance traffic aware data reporting algorithm for object tracking in multi-sink wireless sensor networks. *Wireless Networks*, 24(3), 735–753.

3. Kuila, P., & Jana, P. K. (2017). *Clustering and routing algorithms for wireless sensor networks: Energy efficient approaches* (1st ed.). CRC Press (Taylor & Francis Group). ISBN-13: 978-1498753821.
4. Gupta, S. K., Kuila, P., & Jana, P. K. (2014). E³BFT: Energy efficient and energy balanced fault tolerance clustering in wireless sensor networks. In *IC3I 2014, IEEE Xplore* (pp. 714–719).
5. Rault, T., Bouabdallah, A., & Challal, Y. (2014). Energy efficiency in wireless sensor networks: A top-down survey. *Computer Networks*, 67, 104–122.
6. Muduli, L., Mishra, D. P., & Jana, P. K. (2018). Application of wireless sensor network for environmental monitoring in underground coal mines: A systematic review. *Journal of Network and Computer Applications*, 106, 48–67.
7. Azharuddin, Md., Kuila, P., & Jana, P. K. (2015). Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks. *Computers & Electrical Engineering*, 41, 177–190. (Elsevier).
8. Kuila, P., & Jana, P. K. (2012). An energy balanced distributed clustering and routing algorithm for wireless sensor networks. In *PDGC 2012, IEEE Xplore* (pp. 220–225).
9. Azharuddin, Md., Kuila, P., & Jana, P. K. (2013). A distributed fault-tolerant clustering algorithm for wireless sensor networks. In *2nd ICACCI 2013, IEEE Xplore* (pp. 997–1002).
10. Rebai, M., Snoussi, H., Hnaïen, F., & Khoukhi, L. (2015). Sensor deployment optimization methods to achieve both coverage and connectivity in wireless sensor networks. *Computers & Operations Research*, 59, 11–21.
11. Harizan, S., & Kuila, P. (2019). Coverage and connectivity aware energy efficient scheduling in target based wireless sensor networks: An improved genetic algorithm based approach. *Wireless Networks*, 25(4), 1995–2011.
12. Karatas, M. (2018). Optimal deployment of heterogeneous sensor networks for a hybrid point and barrier coverage application. *Computer Networks*, 132, 129–144.
13. Gupta, S. K., Kuila, P., & Jana, P. K. (2016). Genetic algorithm for k -connected relay node placement in wireless sensor networks. In *Proceedings of the Second International Conference on Computer and Communication Technologies* (Vol. 379, pp. 721–729). In *AISC*. Springer.
14. Gupta, S. K., Kuila, P., & Jana, P. K. (2016). Genetic algorithm approach for k -coverage and m -connected node placement in target based wireless sensor networks. *Computers & Electrical Engineering*, 56, 544–556.
15. Dey, N., Ashour, A. S., Shi, F., Fong, S. J., & Sherratt, R. S. (2017). Developing residential wireless sensor networks for ECG healthcare monitoring. *IEEE Transactions on Consumer Electronics*, 63(4), 442–449.
16. Cardei, I., & Cardei, M. (2008). Energy-efficient connected-coverage in wireless sensor networks. *International Journal of Sensor Networks*, 3(3), 201–210.
17. Rout, M., & Roy, R. (2017). Optimal wireless sensor network information coverage using particle swarm optimisation method. *International Journal of Electronics Letters*, 5(4), 491–499.
18. Sharma, D., & Gupta, V. (2017). Improving coverage and connectivity using harmony search algorithm in wireless sensor network. In *International Conference on Emerging Trends in Computing and Communication Technologies (ICETCCT)* (pp. 1–7). IEEE.
19. Zungeru, A. M., Ang, L. M., & Seng, K. P. (2012). Classical and swarm intelligence based routing protocols for wireless sensor networks: A survey and comparison. *Journal of Network and Computer Applications*, 35(5), 1508–1536.
20. Kuila, P., & Jana, P. K. (2016). Evolutionary computing approaches for clustering and routing in wireless sensor networks. In *Handbook of research on natural computing for optimization problems* (pp 246–266). IGI Global. ISBN: 9781522500582.
21. Deif, D. S., & Gadallah, Y. (2017). An ant colony optimization approach for the deployment of reliable wireless sensor networks. *IEEE Access*, 5, 10744–10756.
22. Gupta, S. K., Kuila, P., & Jana, P. K. (2013). Delay constraint energy efficient routing using multi-objective genetic algorithm in wireless sensor networks. In *ICECCS 2013* (pp. 50–59). Tata McGraw-Hill.
23. Binh, H. T. T., & Dey, N. (2018). *Soft computing in wireless sensor networks*. CRC Press.

24. Binh, H. T. T., Hanh, N. T., & Dey, N. (2018). Improved cuckoo search and chaotic flower pollination optimization algorithm for maximizing area coverage in wireless sensor networks. *Neural Computing and Applications*, 30(7), 2305–2317.
25. Gupta, S. K., Kuila, P., & Jana, P. K. (2016). Energy efficient multipath routing for wireless sensor networks: A genetic algorithm approach. In *5th ICACCI 2016*, IEEE Xplore (pp. 1735–1740).
26. Qasim, T., Mujahid, A., Bhatti, N. A., Mushtaq, M., Saleem, K., Mahmood, H., et al. (2018). ACO-Discreet: An efficient node deployment approach in wireless sensor networks. In *Information technology-new generations* (pp. 43–48). Springer.
27. Xu, Y., Ding, O., Qu, R., & Li, K. (2018). Hybrid multi-objective evolutionary algorithms based on decomposition for wireless sensor network coverage optimization. *Applied Soft Computing*, 68, 268–282.
28. El-Sherif, M., Fahmy, Y., & Kamal, H. (2018). Lifetime maximization of disjoint wireless sensor networks using multiobjective genetic algorithm. *IET Wireless Sensor Systems*, 8(5), 200–207.
29. Wang, J., Ju, C., Gao, Y., Sangaiah, A. K., & Kim, G. J. (2018). A PSO based energy efficient coverage control algorithm for wireless sensor networks. *Computers, Materials & Continua*, 56(3), 433–446.
30. Golberg, D. E. (1989). *Genetic algorithms in search, optimization, and machine learning* (Vol. 1989, No. 102, p. 36). Addison Wesley.
31. Kuila, P., Gupta, S. K., & Jana, P. K. (2013). A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. *Swarm and Evolutionary Computation*, 12, 48–56. (Elsevier).
32. Gupta, S. K., Kuila, P., & Jana, P. K. (2013). GAR: An energy efficient GA-based routing for wireless sensor networks. In *LNCS* (Vol. 7753, pp. 267–277). Springer.
33. Gupta, S. K., Kuila, P., & Jana, P. K. (2016). GA based energy efficient and balanced routing in k-connected wireless sensor networks (Vol. 458, pp. 679–686). In *AISC*. Springer.
34. Bose, A., Biswas, T., & Kuila, P. (2019). A novel genetic algorithm based scheduling for multi-core systems (Vol. 851, pp. 45–54). In *AISC*. Springer.
35. Kennedy, J. (2010). Particle swarm optimization. *Encyclopedia of Machine Learning*, 760–766.
36. Panag, T. S., & Dhillon, J. S. (2018). A novel random transition based PSO algorithm to maximize the lifetime of wireless sensor networks. *Wireless Personal Communications*, 98(2), 2261–2290.
37. Storn, R. (1995). Differential evolution—A simple and efficient adaptive scheme for global optimization over continuous spaces. Technical report, International Computer Science Institute, 11.
38. Kuila, P., & Jana, P. K. (2014). A novel differential evolution based clustering algorithm for wireless sensor networks. *Applied Soft Computing*, 25, 414–425. (Elsevier).
39. Qin, N. N., & Chen, J. L. (2018). An area coverage algorithm for wireless sensor networks based on differential evolution. *International Journal of Distributed Sensor Networks*, 14(8), 1550147718796734.
40. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, 179(13), 2232–2248.
41. Gupta, S. K., Kuila, P., & Jana, P. K. (2016). Energy efficient routing algorithm for wireless sensor networks: A distributed approach. In *Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing Systems (ICCCS 2016)* (pp. 207–213) CRC Press, Taylor & Francis Group.
42. Singh, D., Kuila, P., & Jana, P. K. (2014). A distributed energy efficient and energy balanced routing algorithm for wireless sensor networks. In *3rd ICACCI 2014*, IEEE Xplore (pp. 1657–1663).
43. Kuila, P., & Jana, P. K. (2014). Approximation schemes for load balanced clustering in wireless sensor networks. *Journal of Supercomputing*, 68, 87–105. (Springer).
44. Kuila, P., & Jana, P. K. (2014). Heap and parameter based load balanced clustering algorithms for wireless sensor networks. *International Journal of Communication Networks and Distributed Systems*, 14(4), 413–432.

45. Kuila, P., & Jana, P. K. (2012). Improved load balanced clustering algorithm for wireless sensor networks. In *LNCS* (Vol. 7135, pp. 399–404). Springer.
46. Kuila, P., & Jana, P. K. (2012). Energy efficient load-balanced clustering algorithm for wireless sensor networks. *Procedia Technology*, 6, 771–777. (Elsevier).
47. Konak, A., Coit, D. W., & Smith, A. E. (2006). Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9), 992–1007.