Wireless Netw (2018) 24:683–693
https://doi.org/10.1007/s11276-016-1361-5

# Performance evaluation of sensor deployment using optimization techniques and scheduling approach for K-coverage in WSNs

Muralitharan Krishnan[1] · Vishnuvarthan Rajagopal[2] · Sakthivel Rathinasamy[3]

Published online: 1 September 2016

**Abstract** In the wireless sensor networks, sensor deployment and coverage are the vital parameter that impacts the network lifetime. Network lifetime can be increased by optimal placement of sensor nodes and optimizing the coverage with the scheduling approach. For sensor deployment, heuristic algorithm is proposed which automatically adjusts the sensing range with overlapping sensing area without affecting the high degree of coverage. In order to demonstrate the network lifetime, we propose a new heuristic algorithm for scheduling which increases the network lifetime in the wireless sensor network. Further, the proposed heuristic algorithm is compared with the existing algorithms such as ant colony optimization, artificial bee colony algorithm and particle swarm optimization. The result reveals that the proposed heuristic algorithm with adjustable sensing range for sensor deployment and scheduling algorithm significantly increases the network lifetime.

**Keywords** Scheduling · Sensing range · Sensor deployment · Target coverage · Wireless sensor networks

✉ Sakthivel Rathinasamy
krsakthivel@yahoo.com

Muralitharan Krishnan
muralitharank.ooty@gmail.com

Vishnuvarthan Rajagopal
rvvarthan@gmail.com

[1] Department of Mathematics, Sri Ramakrishna Institute of Technology, Coimbatore 641 010, India

[2] Department of Computer Science, Anna University, Regional-Centre, Coimbatore 641 047, India

[3] Department of Mathematics, Sungkyunkwan University, Suwon 440-746, Republic of Korea

## 1 Introduction

A sensor is a small, low cost and low power operating device which is capable of sensing, processing and transmitting the sensed data through wireless network. Wireless Sensor Networks (WSN) plays a vital role in many applications such as security, battlefield surveillance, air traffic control, biological detection, environment monitoring, industrial automation, smart grid etc., [1, 2]. To monitor such applications, the environment requires a bunch of sensors to be deployed. The area that the sensor can monitor or sense is called as coverage of the sensor [3]. A sensor can monitor one or more targets within its sensing range. Due to the small size of the sensor and deploying it in inaccessible regions, replacing or charging of the battery is impractical [4]. Hence, increasing the network lifetime is one of the significant issues in WSN. Optimal deployment of sensors [5–7], sleep scheduling (i.e.) switching the sensors between active and sleep modes [8–11], and load balancing [12] are the major factors that must be considered to increase the network lifetime.

In general, sensor node deployment can be categorized into two types: random deployment and deterministic deployment [13, 14]. Random deployment is highly suitable for the unknown region or inaccessible region. The drawback in random deployment or un-deterministic deployment is that the placed node may or may not cover the target. The nodes which are not deployed near the target might not able be to capture the useful information. In order to overcome this problem, the node must be moved towards the target after deployment. Since the region is inaccessible the node must have the ability to move by itself. Whereas in the deterministic deployment, regions are well known to place the sensors. In such scenarios, the sensor network lifetime can be extended either in the deployment phase or in scheduling phase. In deployment phase, the sensor nodes are placed in the optimal locations to increase the network lifetime with guaranteed coverage [15]. To achieve the optimal node placement there are some bio-inspired algorithms, namely, ant colony optimization (ACO), particle swarm optimization (PSO), artificial bee colony (ABC) algorithm, genetic algorithm, etc,.

Sensor coverage can be classified into area coverage and target coverage. If the sensor network covers or senses the entire region, then it is called as area coverage. If the sensor senses the definite target (point) in the specified region, then it is called as target coverage or point coverage [4]. Target coverage can be categorized as simple coverage, k-coverage and Q-coverage. In simple coverage each target is monitored by at least one sensor node. In case of sensor node failure the target remains uncovered. To overcome this problem, in k-coverage the target is monitored by at least 'k' number of sensors. In Q-coverage, a set of targets are monitored by a set of sensor nodes (i.e.) 'n' number of targets may covered by 'q' number of sensors.

The energy available in the sensor nodes are limited and so the WSN devices are called as energy sensitive devices [16]. So, in order to use the energy efficiently, scheduling the sensor nodes to monitor the target is necessary. While scheduling the sensor nodes, network congestion and packet retransmission can be avoided, which drastically reduces the usage of extra energy for communication in the network. Scheduling the sensors for target coverage increases the network lifetime. Instead of monitoring a single target with number of sensors, number of targets can be monitored by limited number of sensors in groups in a round robin scheme [17, 18]. Number of sensors can be grouped into sets where the sets may or may not have duplicates. When a set is active, its element sensor nodes will be in either idle or active mode, while all the other sensors which are the elements of other sets will be in sleep mode.

Load balancing is an other important factor to achieve maximum network lifetime. The available energy of each sensor varies drastically depending on its sensing range, sensor's distance from base station and the amount of transmission power consumed by the sensor. Depending on the available residual energy, the load can be balanced between the sensors.

The main contribution of this paper are summarized as follows:

- A new heuristic algorithm is proposed for sensor deployment with automatic adjustment of sensing radius.
- Optimization algorithms are employed for performance analysis on the network lifetime of WSN.
- A novel sensor scheduling algorithm is designed for increasing the lifetime of the WSN with a minimum dominating sets approach.
- Performance analysis to ensure target monitoring, network connectivity and quality of service in WSNs.
- Simulation results are presented to demonstrate the efficiency of the proposed technique over the existing optimization techniques.

The rest of the paper is organized as follows: In Sect. 2, the related works are discussed. Section 3 explains the proposed heuristic algorithms for node deployment and optimization techniques for increasing the network lifetime. In Sect. 4, sensor scheduling mechanisms are discussed. Section 5 showcases the results and conclusion is presented in Sect. 6.

## 2 Related work

In WSNs, sensing and target coverage are important metrics of quality of service (QoS). Using sensing range slicing and active node scheduling methodologies, the authors in [9] dealt with the problem of k-coverage in homogeneous and heterogeneous WSNs. Two k-coverage protocols (1) self scheduling driven k-coverage and (2) triggered-scheduling driven k-coverage are proposed with their performance analysis in this work. However, realistic sensing model is not considered in [9].

In [19], the authors have used hybrid PSO and differential evolutionary algorithm to solve the target coverage problem. In WSNs, a node monitors an event when it occurs nearer to it. If the event occurs far away from the node, then the detection accuracy will be less. [20] presents localized coverage quality algorithm (LoCQAl) to satisfy the detection accuracy which is based on the mobile nodes. To extend the network lifetime, the monitoring process is divided into rounds. In each round, only a set of sensors will be active while the rest of the nodes would remain in sleep mode to conserve its energy.

Zorbas et al. [23] deals with energy-efficient target detection in WSNs with random node deployment. The work evaluates the probability of missed detection of one or more incoming targets, delay for detecting a target, latency of notification transmission and network lifetime. But [23] does not deal with the deterministic node deployment.

Zhao et al. [24] splits the deployed sensors into number of sensor sets or groups. Each set monitors the target and sends the data to sink. The sensor sets are activated sequentially. Only the sensors which are in active state senses the targets and relays the data to the sink. All the other sensors will enter into energy-saving sleep mode. The authors in [24] developed a heuristic algorithm called communication weighted greedy cover (CWGC), which uses a greedy method to select a set of source nodes to cover the target. The algorithm couples the communication cost and the selection of source sets.

Target coverage problem in wireless heterogeneous sensor networks (WHSNs) is considered in [4]. There are two heuristic distributed schemes remaining energy first scheme (REFS) and energy efficient first scheme (EEFS) to reduce the target coverage problem. REFS considers remaining energy in each sensor and neighbors decisions to enable its sensing units to ensure every target being covered. EEFS considers its sensing capabilities and remaining energy of its neighbors to make a better decision to turn on its sensing units and to ensure each target being covered.

In WSN, the sensor must be deployed appropriately to achieve better system performance. Kim et al. [25] proposes an effective sensor deployment scheme based on the event occurrence rate in a particular region. With the help of this scheme, better surveillance of the targets can be achieved with the minimum number of sensors. If a region experiences an imperfect event occurrence, then this scheme would fail to deploy the sensors in an effective way.

Mixed-integer linear programming model is developed in [26] to maximize the network lifetime. The proposed method determines the optimal locations for the sensors. It also schedules the work activity for deployed nodes. When there is a high data packet generation rate, sensor network lifetime would get decreased significantly.

The fundamental problem in WSN is minimum latency data aggregation schedule. It can be solved by adjusting the transmission radii of the sensors. Hong et al. [27] partitions the set of sensor nodes into major and minor sets. The minor set aggregates the data to the major set and the major set aggregates the data to the base station [27].

In smart grid, the number of active sensors are required to satisfy the service requests. The main challenge here is to handle the limited battery energy of sensor node efficiently. To solve this problem, sleep scheduling model is proposed based on round robin in [28]. Energy efficient scheduling is considered to minimize the transmission energy cost. Multi-access scheduling problem is reformulated into job scheduling problem and it is solved by adopting job scheduling policies [29].

### 2.1 Preliminaries and definitions

In this section, we have few definitions that help to build a wireless sensor network.

1. *Sensing radius* $(s_r)$ In the given region $R$, a sensor $s_i$ is placed. $s_i$ can cover the region ($R$) from its location $s_i$(x,y).
2. *Communication range* $(C_r)$ Sensor $s_i$ can communicate with a Target $t_j$, a sensor node $s_n$ or a base station (BS) in the given region R.
3. *Connectivity* ($C$) Sensor nodes can communicate with other nodes. The packet from a node can be transmitted to base station either directly or indirectly through intermediate nodes.
4. *Degree of coverage* $(D_c)$ In a given region $R$, the ability of the sensor $s_i$ to monitor the average number of targets.
5. *Simple coverage (c)* In the given region R, a target $t_j$ is monitored by at least one sensor $s_i$, where $s_i, t_j \in R$. i.e., $D_c = 1$.
6. *K Coverage (k)* In the given region R, a target $t_j$ is monitored by sensor $s_i$, where $s_i$, $t_j \in R$. i.e., $D_c = k$. $t_j$ is monitored by at least k sensor nodes, where k is a predefined constant and all k number of nodes must be in active state.
7. *Threshold limit (T)* The number of targets to be monitored by a sensor $s_i$ can be limited. Base station can be able to set the threshold limit T for Sensor $s_i$.

### 2.2 Upper bound lifetime of a sensor networks

In a given region R, i sensors $s_1, s_2, \ldots s_i$ are deployed to cover j targets $t_1, t_2, \ldots tj$. The initial power of sensor node is $b_i$. To check the connectivity, it is assumed that the sensor can compute the distance of neighbor node and the target to be monitored. The Euclidean distance [13, 28] between two points (sensor (x,y) and target (a,b)) within the region R is defined as

$$E_d(s,t) = \sqrt{(x-a)^2 + (y-b)^2}, \tag{1}$$

where sensor coordinate points are denoted by (x,y) and (a,b) denotes target coordinate points. To find the distance between the two objects (sensor and target), Eq. 1 is used. The distance between a sensor and a target (geometric object) is like a line in the plane with the minimal distance $E_d(s,t)$. If the distance between the sensor $s_i$ and target $t_j$ is less than the sensing radius $s_r$, then the coverage matrix $C_m(i,j)$ can be given by

$$C_m(i,j) = \begin{cases} 1, & if \;\; E_d \;\; \leq \;\; s_r \\ 0, & otherwise \end{cases} \tag{2}$$

The lifetime of the sensor is the ratio of the initial battery power and energy consumption rate. Lifetime of a sensor battery $(l_b)$ can be denoted as

$$l_b(i) = \frac{b_i}{e_i}, \tag{3}$$

where $b_i$ represents the initial battery power and $e_i$ denotes the energy consumption rate of a sensor $s_i$. $C_m$ matrix contains the information about the wireless sensor network coverage. The elements of the matrix $C_m$ are 1 and 0. If the target is covered by the particular sensor, then the value of the matrix element $C_m[i.j]$ is 1 else it is 0. The product of coverage matrix $C_m$ and life time of battery $l_b(i)$ yields the network lifetime of the WSN [14, 21, 22]. The theoretical calculation of upper bound for network lifetime can be calculated using Eqs. 2 and 3 as

$$N_{lifetime} = \frac{\sum_{i=1}^{n} C_m(i,j) * l_b(i)}{k_j}, \tag{4}$$

where $k_j$ represents k-coverage with, $j = 1, 2, 3, \ldots n$.

### 2.3 Degree of coverage

The average number of targets that a sensor can monitor in a region $R$ is termed as degree of coverage $D_c$. The node density can be indicated with the help of $D_c$. In a fixed area, when the number of sensors increases, the coverage of the area also increases. Fine grained spatial information can be gathered with high node density. In general, degree of coverage is classified into higher degree and lower degree. Higher degree sensor node are costlier devices which has high sensing range and provides more reliable communication between the devices. While lower degree of a sensor node has low sensing range which are cheaper devices and provides desirable coverage degree. But it entirely depends upon the application and the event characteristics.

For better understanding the sensor deployment problem with the target coverage, the following bio-inspired and evolutionary existing algorithms are discussed in detail:

- Artificial bee colony algorithm
- Ant colony optimization algorithm
- Particle swarm optimization algorithm

The above mentioned algorithms are deeply analyzed with their corresponding pseudo codes in Sects. 2.4, 2.5 and 2.6 respectively.

### 2.4 ABC algorithm for sensor deployment

Using the ABC algorithm, maximizing the lifetime of the sensor by optimal placement can be achieved [30, 31]. ABC is an optimization technique based on the behavior of honey bee swarm. The bee colony consists of employer bees, onlookers and scouts. The employer bee loads its nectar, arrives to hive from source. After unloading it, employer bees perform waggle dance which contains the information about the direction, distance and the quality of food source [32].

Set the control parameters $L$ and $C_s$. Limit $L$ for scout bee is calculated by

$$\mathbf{L} = (\mathbf{C_s} * \mathbf{D})/2, \tag{5}$$

where $C_s$ represents the colony size and $D$ represents the dimensions of the function.

Initialize the position of the targets randomly and evaluate the objective function using the Eq. 4. To monitor the selected target, the minimum distance from sensor to target is the best fitness value of the sensor for deployment. To obtain a new food source, the following formula is used:

$$v_{i,j} = x_{i,j} + \Phi_{ij}(x_{i,j} - x_{k,j}), \tag{6}$$

where $\Phi_{ij}(x_{i,j} - x_{k,j})$ is a random number in the range [−1, 1]. Compare the obtained new food source information with the previous food source. If the new solution is better than the old one, then replace the old food source with the new one. Calculate the probability value $P_i$ for the obtained solutions based on the fitness values, by using

$$P_i = \frac{f(i)}{\sum_{i=1}^{m} f(i)}, \tag{7}$$

where $m$ is the total number of targets.

The onlooker finds a neighborhood target which is in the vicinity of $D_i$, by using the following equation

$$D_i(t+1) = D_i(t) + \delta_{ij} * r, \tag{8}$$

where $\delta_{ij}$ is the neighborhood patch size for j-th dimension of i-th target and $r$ is a random uniform variate in the range [−1, 1].

Evaluate new solutions for the onlooker by using the objective function. If the new solution is better, then onlooker memorizes it and discards the previous solution. Repeat the above process until the stop condition is met.

The Algorithm 1 describes the ABC algorithm. When the distance of transferring the information increases, then the lifetime of the sensor decreases. But the ABC algorithm

**Algorithm 1** ABC Algorithm for Sensor Deployment

```
{$S_i$: Sensor set }
{$T_s$: Set of targets covered by a sensor }
{new Sol: Solution from the current iteration }
{prvSol: Solution from the previous iteration }
{$noi$: number of iteration}
{$maxIter$: maximum iteration}
Initialize population
Evaluate fitness
do
    Form new population
    Search for the new solution in the neighborhood
    if ($newSol$ is better than $prvSol$) then
        Store the new solution
        Forget the previous solution
    else
        Keep the previous solution as it is
    end if
    $noi = noi + 1$
while ($noi = maxIter$)
```

places the sensors optimally in such a way that the distance between the sensor, target and the base station is small. But sometimes, finding the new solution using some random number might not effectively yield a optimum solution.

### 2.5 ACO algorithm for sensor deployment

Ant colony optimization (ACO) is one of the suitable algorithm for finding the optimal solutions for the broad range of combinatorial problems. A group of ants communicate to find a good solution by depositing the pheromone on its way to food source, while searching for it from their nest [33–35]. The path selection depends only on the intensity of the pheromone, which was deposited by the ants during the food search.

**Algorithm 2** ACO Algorithm for Sensor Deployment

```
{$S_i$: Set of sensors }
{$T_i$: Set of targets }
Initialize: Control parameters (Design variable, pheromone range, number of ants)
Initialize: $S_i$, $T_i$, Pheromone
do
    Calculation of Prob path
    Path construction
    Random generation of ants
    Path identification
    Evaluate the fitness (objective function)
    Selection of optimum values
    Update pheromone values (local and global)
    Update probability values
    $noi = noi + 1$
while ($noi = maxIter$)
```

Calculate the distance between the sensor and the target using the Euclidean values d(i,j). Find the best feasible location to place the sensor for monitoring the targets. The process is recursively performed until all the sensors find its best feasible position in the given region. To select the next target to be monitored by a sensor, the sensor must select the target based on density of the pheromone which has the next highest probability.

$$P_{ij} = \frac{\tau_{ij}}{\sum_{p=1}^{n} \tau_{ip}}, \tag{9}$$

where $P_{ij}$ denotes the probability of finding the food source by an ant, $\tau_{ij}$ is pheromone initialization matrix value, $\tau_{ip}$ is the total number of pheromone in the pheromone initialization matrix.

The distance between the node and the target is computed and stored. Local pheromone matrix is computed by

$$pherom(j,p) = (1-\rho) * P_{pherom}(j,p) + (rand_1 * min1), \tag{10}$$

where *pherom*(*j*, *p*) represents the local pheromone matrix value, $\rho$ is the control parameter and $rand_1$ is a random number.

The global pheromone matrix is computed by

$$\begin{aligned} phrom(f,i) = (1-\rho) * pherom(f,i) + (rand_2 \\ + (1/(f_{best}+1))), \end{aligned} \tag{11}$$

where *phrom*(*f*, *i*) represents the global pheromone matrix value, *fbest* denotes the feasible best value and $rand_2$ is a random number.

### 2.6 PSO algorithm for sensor deployment

Particle swarm optimization (PSO) algorithm is based on the social behavior of a group of birds [13]. The birds or particles moves in the search space to find the optimal food source. Typically, the birds would find food by random in the hyperspace. Then the information from the different particles about the food sources are collected and evaluated based on the objective function. The fitness cost of a bird that is closer to the global solution is lower than the particle which is farther. Velocity V and position X are updated in each iteration and the update process is repeated until the global best is achieved. After finding the best solution, it is communicated among the particles.

**Algorithm 3** PSO Algorithm for Sensor Deployment

```
{$S_i$: Set of sensors }
{$T_s$: Set of targets monitored by a sensor s }
{$noi$: Number of iteration}
{$maxIter$: Maximum number of iteration}
Initialize the particles
do
    loop $i := 1$ to $n$
        Evaluate the fitness
        Identify the local and global best values
        if ($pbest > gbest$) then
            Store the new solution (pbest)
            Remove the old solution (gbest)
        end if
    end loop
    loop $j := 1$ to $n$
        Update the velocity
        Update the Position
        Update pbest and gbest of the particles
    end loop
    $noi = noi + 1$
while ($noi = maxIter$)
```

Initially, the sensor positions are randomly generated and the distance between the sensor and the target cost is stored in (pbest). The global best sensor locations are computed and stored in (gbest). The velocity and the sensor locations are adjusted according to bird's own flying experience. The velocity V and the position s are updated at each iteration using

$$\begin{aligned} V_i^{k+1} =& w * V_i^k + c_1 * rand_1 * (pbest_i - s_i^k) \\ &+ c_2 * rand_2 * (gbest - s_i^k), \end{aligned} \tag{12}$$

$$s_i^{k+1} = s_i^k + V_i^{k+1}, \tag{13}$$

where $w$ is the weighting function; $c_1$ and $c_2$ are the constants which denotes the weighting factor; $rand_1$, $rand_2$ generates random numbers between 0 and 1; pbest is the personal best value while gbest refers to global best value; $V_i^k$, denotes the velocity of particle $i$ at the iteration $k$; $s_i^k$, represents the position of the particle $i$ at the iteration $k$.

## 3 Description of the deployment heuristic algorithm

All the sensor nodes are randomly deployed in the given region R. Network lifetime is computed by considering that initially every node has equal amount of energy (battery) and equal sensing range. Using proposed heuristic algorithm, sensing radius adjustment is performed in the sensor network by assuming every sensor has the tendency to automatically adjust the sensing range ($s_r$) with the help of a base station. Base station(BS) also called as data collector node, is responsible for scheduling activities and assume that the BS is stuffed with unlimited energy, bandwidth and coverage in the network. For performance evaluation, the sensors are deployed with heuristic algorithm [14], ABC algorithm, ACO algorithm and PSO algorithm. Finally, the upper bound lifetime of the network is computed before and after the scheduling the sensors.

### 3.1 Deployment heuristic algorithm

**Algorithm 4** Deployment Heuristic Algorithm

```
1: {$S$: Set of sensors }
2: {$S_i$: Number of Sensors in the set S }
3: {$T_s$: Maximum number of targets covered by a sensor $s_i$ }
4: {$T_h$: Threshold value or maximum number of targets that a sensor $s_i$ can monitor }
5: {$s_r$: Sensing radius of a sensor}
6: {$T_d$: Distance of target from sensor }
7: {$D_m$: Max distance of target from sensor $s_i$}
8: Deploy the sensors randomly
9: Compute coverage
10: do
11:     if ($T_s$=$T_h$) and ($T_d <= s_r$) then
12:         $s_r = D_m$
13:         Perform connectivity check
14:         Ensure the monitoring
15:         Update coverage
16:     else
17:         if ($T_s < T_h$) then
18:             Place sensor to monitor a maximum number of target
19:             Shift the sensor to center of the targets
20:             Repeat Step 11 to 15
21:         else
22:             Monitor the targets
23:         end if
24:     end if
25: while ($maxIter = noSen$) and ($Si\ covers\ at\ least\ single\ target$)
```

The deployment heuristic Algorithm 4 places the sensors randomly in the given region. Threshold value for the sensors is set to cover the number of targets. Each sensor can monitor more than one target. The Algorithm 4 computes the upper bound lifetime of the network. If the number of targets covered by the sensor equals the threshold value, then the sensor $s_i$ calculates the coverage distance from i to j for the monitored targets and stores the values. If the distance between the sensor and the monitored targets is less than sensing radius ($s_r$) then the sensor ($s_i$) can adjust its sensing radius according to the distance value of monitored target ($t_j$).

$$S_r'(i) = S_r(i) - q, \tag{14}$$

where $s_r'(i)$ is the new sensing radius of node i, $s_r(i)$ is the actual sensing radius of node i and q is the constant distance which is to be reduced from actual sensing radius. Because of shrinking the sensing radius of the node, power consumption rate of the sensor mode will also get reduced. Proportionately, it will increase the lifetime of the network. Now the lifetime of the battery ($l_b'$) can be calculated as

$$l_b'(i) = \frac{b_i}{e_i'}, \tag{15}$$

where $b_i$ denotes initial battery power and $e_i'$ denotes the current energy consumption rate of sensor $s_i$.

So the new network lifetime can be recalculated as

$$N_{lifetime}' = \frac{\sum_{i=1}^{n} C_{ij} * l_b'(i)}{k_j}. \tag{16}$$

The new network lifetime obtained using Eq. 16 will be definitely higher than the initially calculated network lifetime which is obtained through Eq. 4. The connectivity check and target monitoring check must be performed periodically to ensure the liveliness of WSN. If a sensor could not monitor any of the targets, then the sensor is moved to a new location where it can monitor the targets up to its threshold limit.

## 4 WSN scheduling

Node scheduling is the second part of the proposed work. Heuristic algorithm is proposed using the minimum dominating sets approach and the numbers of sensors are divided into a number of groups based on this approach. Each group has a number of sensors which ensures the highest degree of network coverage. Round robin based scheduling mechanism is implemented in [17] that ensures the maximum obtainable network lifetime in the sensor network.

Thus, after efficient deployment of the sensors, residual energy of the each sensor is calculated. These sensors are grouped together based on the target that it is monitoring. After grouping and ordering the sensor in descending order, based on the residual energy, sets are formed and time span for sensing the targets for a set is decided. In order to balance the energy and increase the lifetime of the network sensor set switching is performed. Heuristic scheduling algorithm is segmented into 4 parts. (A) Residual energy calculation (B) Grouping sensors (C) Set formation and (D) Sensor set switching.

**Algorithm 5** Scheduling Algorithm

```
  I: Residual Energy Calculation
 II: Grouping Sensors
III: Set Formation
 IV: Sensor Set Switching
```

### 4.1 Residual energy calculation

The usable energy available in the sensor is called residual energy. Sensor can work in three modes. (1) Active mode (2) Idle mode and (3) Sleep mode. In active mode, sensor transmits and receives the data, to and from the base station with its radio on. Timer ticks continuously in all three modes. In idle mode, the radio is set to idle, where no communication with other sensors would take place. In sleep mode, the whole component will be in off mode, except the timer. The maximum energy can be saved in these mode. Initially, the residual energy of the sensor $E_{ri}$ is the same as the total maximum energy $E_{max}$ available at the sensor. The total energy consumed by a sensor, $E_c$, can be given as the sum of energy consumed by the sensor in active mode, idle mode and sleep mode. Therefore, the difference between $E_{max}$ and total energy consumed will be the residual energy of the sensor.

**I:** Residual Energy Calculation

```
{$s_i$: Sensor i }
{$t_i$: Target i }
{$E_{max}i$: Initial maximum energy available at $s_i$ }
{$E_ri$: Residual energy of $s_i$ }
{$E_c$: Energy consumed by $s_i$ }
{$a_i$: Energy consumption of $s_i$ in active mode }
{$i_i$: Energy consumption of $s_i$ in idle mode }
{$sl_i$: Energy consumption of $s_i$ in sleep mode }
$E_ri$ = $E_{max}i$
loop $i := 1$ to $n$
    $E_c$= $a_i$ + $i_i$ + $sl_i$
    $E_ri$ = $E_ri$ - $E_c$
    Send ($s_i$, $E_ri$) values to base station
end loop
```

### 4.2 Grouping the sensors

For M targets, there may be N sensors deployed in the field. A sensor can sense one or more targets depending on its sensing range and the number of targets available within that range. A sensor set is formed for every target. If a sensor can sense more than one target, then the sensor will be added to all the target groups. That is, if a sensor can sense 3 targets, then the sensor will be in 3 groups. The only set, which is in active mode, is responsible to cover the entire region R, while the other sets are set to sleep mode.

**II:** Grouping the Sensors

```
{$T_i$: Target sensor group/set for the target $t_i$ }
loop $i := 1$ to $n$
    loop$j := 1$ to $n$
        if Sensor $sj$ Senses $ti$ then
            Add $sj$ to $Ti$ Group
        end if
    end loop
    Order the sensor based on residual energy in decreasing order
end loop
```

### 4.3 Set formation

Set is formed in such a way that it increases the lifetime of the network. Considering $k$ coverage, each target should have $k$ sensors to monitor it. For n targets in a region, a sensor set $S_i$ should have $k$ to $k*n$ sensors where $k = 1, 2, 3, \ldots n$. If a set $S_i$ comes into active mode, all the sensors in the set, senses the target, transmits and receives message between sensors and base station. Each sensor set will gets its turn in round robin basis. The other sensors which does not belong to the active set, must wait for its turn in sleep mode.

The sensors must be ordered in descending order of residual energy before selecting the sensors for the sensor set $S_i$ from the target group $T_i$. For each target, $k$ sensors are selected in first in first out basis from the target group. So, the set-1 should have high average residual energy than the set n. If number of sensor that can sense the target $t_i$ available in set $S_i$ is equal to $k$, then the selection of sensors for the next target is performed for the same set $S_i$. Else if the number of sensor ($N_s$) is not equal to $k$ and can monitor the target $t_i$, then it must select k - $N_s$ number of sensors from the target group $T_i$. By repeating the same process, n number of sets are formed.

**III:** Set Formation

```
{$S_i$: Set of sensors that can cover the entire region or every target}
{$T_i$: Target sensor group/set for the target $t_i$ }
{$S_n$: Number of sensors that can cover the target $t_i$ in the set $S_i$ }
do
    Find the number of sensors $S_n$ in the set $S_i$ that can sense $t_i$
    if ($S_n$ = $k$) then
        Move to the next target
    else if ($S_n < k$) then
        Select $k - S_n$ sensors for the sensor set $S_i$
        Remove the selected sensors from target group $T_i$
    end if
while (The target groups are non-empty)
```

### 4.4 Sensor set switching

If there is any change or activity in the target, the sensor swifts to active mode to sense the activity and to report it to the base station. If no activity is detected, the sensor would remain in idle mode. In addition to it, to increase the network lifetime of a WSN, it is essential to switch the mode of the sensor between active/idle mode and sleep mode. But the frequent switching between the modes costs more energy than a sensor node in the idle mode. So, frequent switching between the modes must be reduced. The amount of time, a sensor set $S - i$ to be in active/idle mode, is calculated based on the average residual energy of the set $S - i$. Round robin method is used to select a sensor set to be in active/idle mode. Only one sensor set $S_i$ will be in active/idle mode at a time, while the other sets will reside in sleep mode.

Since sensor set-1 has more average residual energy than the sensor set-n, set-1 is allocated with more sensing time. This balances the usage of energy and avoids quick failure of sensors. If any of the sensor set fails, then the set reformation mechanism is mandatory. During initial network formation stage, computation and communication cost must be considered.

The starting time to monitor the target, amount of time the sensor is in active mode and amount of time the sensor to be in sleep mode are computed at the base station and the information is sent to the sensors. Since the computations are performed at the base station which has unlimited power source, the energy at the sensor node can be saved.

**IV:** Sensor Set Switching

```
{$S_i$: Set of sensors that can cover the entire region or every target}
{$A_e i$: Average residual energy of the sensor set $S_i$ }
{$S_t i$: Start time of sensor set $S_i$ to switch to active mode }
{$M_t i$: Monitoring time of a sensor set $S_i$ }
{$Sl_t i$: Sleep time of a sensor set $S_i$ }
{$C$: A constant factor to be multiplied with $M_t i$ to avoid frequent switching between the modes }
Calculate the average residual energy $A_e i$ of every sensor set $S_i$
$S_t(1) = 0$;
if any failure of sensor node reported then
    Perform Set Formation algorithm
else
    loop $i := 1$ to $n$
        $M_t i = (A_e i/A_e n)*C$
        Calculate the sleep time $Sl_t i$ for the sensor set $S_i$
        For every sensor in set $S_i$
            Send($S_t i$,$M_t i$,$Sl_t i$)
        $S_t(i+1) = S_t i + M_t i$ //update the start time of the next sensor set
    end loop
end if
```

## 5 Simulation results

The simulation setup and test is done in MATLAB. All the experiments have been performed with more than 5 to 10 test runs and the results are obtained based on the following configuration parameters:

1. Coverage Region (*R*) 500 m * 500 m.
2. The number of sensors (*S*) in the network is 50–250.
3. The number of targets (*T*) is 25.
4. *K* coverage implementation, where k = 1 to k = 5.
5. Sensing range is initially fixed as 100 m. It can be adjustable after successful deployment with the help of the base station.

### 5.1 Simple coverage

The main objective of the simple coverage is that, every sensor can cover at least one target within the given region. If the sensing range is automatically adjusted by the sensor with the help of a base station, then the lifetime of the sensor can be increased. In general, the random deployment has two drawbacks. First, a target may not be covered by the sensor. Second, uneven deployment of sensors where some targets may be monitored by a number of sensors and some targets do not have any sensor to monitor them. To overcome these drawbacks during the deployment of sensors, optimal deployment technique must be followed.

In Figs. 1, 2, 3, 4, 5, H1 refers to the proposed deployment heuristic, H2 refers to the existing heuristic [14], abc refers to artificial bee colony, aco refers to ant colony optimization, and pso refers to particle swarm optimization. Similarly, the algorithm name with prefix ‘s-’ refers the above mentioned algorithms after scheduling. for instance, s-H1 refers to the proposed heuristic algorithm after scheduling, s-H2 refers to existing heuristic algorithm [14] after scheduling and so on.

After analyzing the Fig. 1, it can be stated that, the proposed heuristic algorithm gives better result, in both cases that is with and without scheduling compared with the existing heuristic [14] and other optimization algorithms for sensor deployment. From the obtained results, it can be concluded that, reducing the sensing range would increase the network lifetime to a certain level in simple coverage.

### 5.2 K coverage

The proposed heuristic approach consistently achieves better results which is shown through Figs. 2, 3, 4, 5 while comparing it with the existing heuristic and other optimization algorithms.

The number of iterations performed for all the algorithms is 1000. With 25 targets and 50–250 number of sensors, 10 test runs were carried out for each algorithm in the simulation setup for analyzing the performance of the algorithms using k-coverage. Random deployment method is used to deploy the sensors for analyzing the performance through Figs. 2, 3, 4, 5 with the k values greater than 2. On successful completion of the deployment, sensors are scheduled to increase the network lifetime further. If the frequency of switching between the active/idle and sleep modes is high, then the total network lifetime is reduced. So, the proposed scheduling algorithm reduces the switching frequency. From the simulation results, it can be concluded that, the proposed scheduling algorithm achieves better network lifetime over the other algorithms.

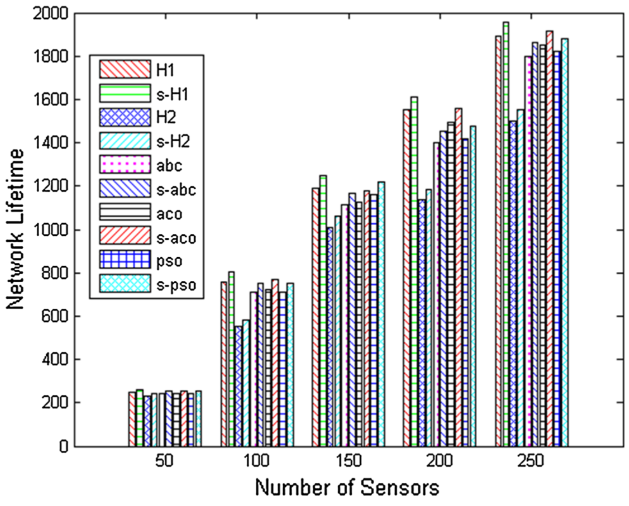

**Fig. 1** Simple coverage

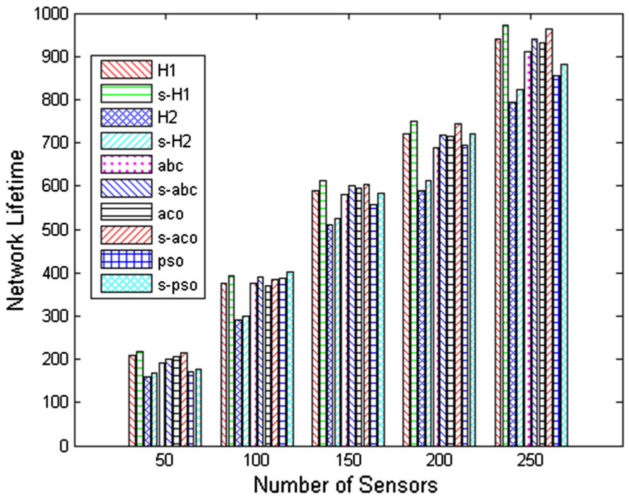

**Fig. 2** K coverage (k = 2)

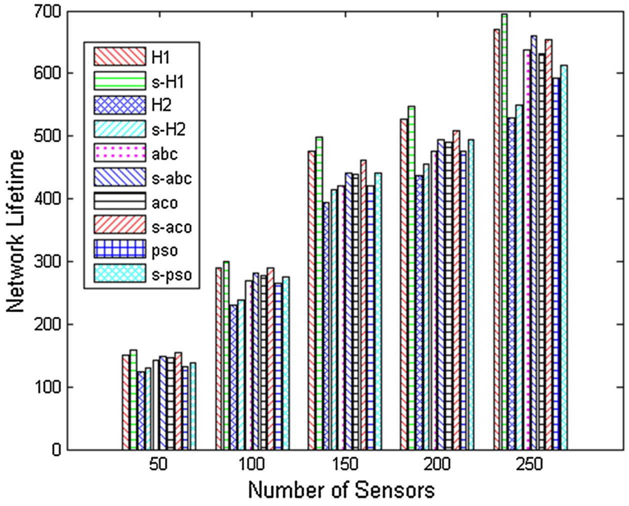

**Fig. 3** K coverage (k = 3)

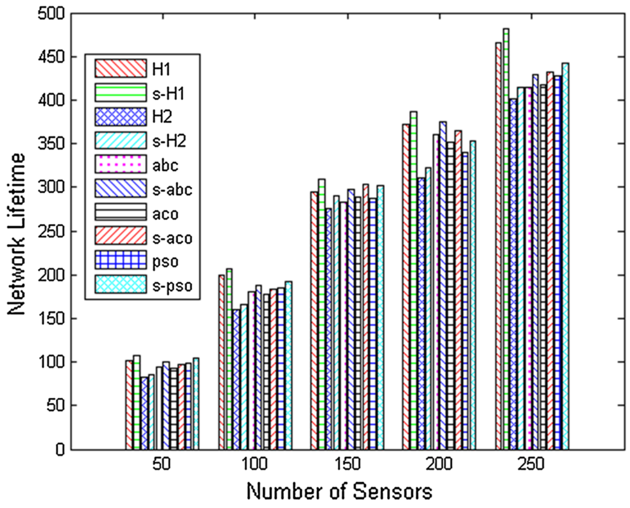

**Fig. 4** K coverage (k = 4)

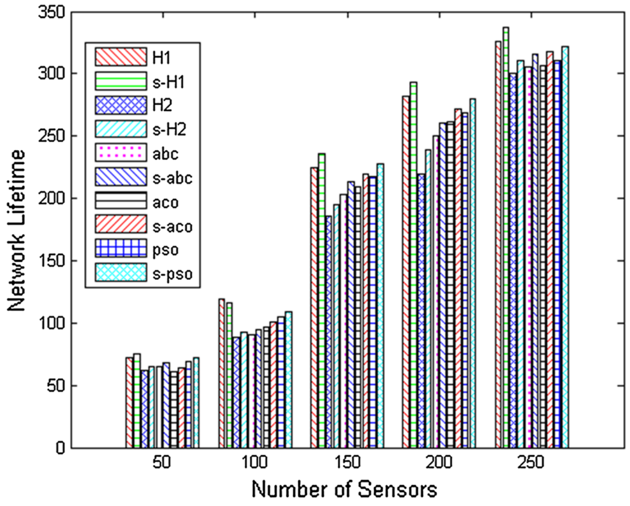

**Fig. 5** K coverage (k = 5)

### 5.3 Observations

*(1) Distance value:* The following scenario explains the effect of the distance between the target and sensor: If the *q* value (distance) is high, then the sensor network lifetime is low and if *q* value is low then the sensor network lifetime is high. Thus *q* is inversely proportional to the sensor network lifetime. For example, if a sensor monitors 2 targets with sensing range 100m:

*Scenario 1.* If both the targets are located far away from the sensor (greater than 90 m and less than 100 m) then the network lifetime is low.

*Scenario 2.* If both the targets are located in equal distance from the sensor (for example, 40–60 m) then the network lifetime is average.

*Scenario 3.* If both the targets are nearer (for example, less than 20 m), then the sensor network lifetime is marginally high due to low energy consumption.

*Scenario 4.* If a target is located nearer (example, less than 20 m) to the sensor and another target is located far away from the sensor (example, greater than 80 m and less than 100 m) then the network lifetime is marginally below average.

From the above scenarios, there is a trade off between the distance $q$ and the network lifetime. Hence $q$ plays an important role in the WSN network lifetime.

*(2) Degree of coverage:* Higher degree devices are deployed which by itself have higher target coverage and network lifetime. By adjusting the sensing radius, network lifetime can be further improved. But for some cost effective applications, higher degree devices cannot be used. In this scenario, to increase the target coverage sensor node density is increased to some extent. Further, the sensing radius is adjusted to improve the network lifetime without adding additional cost.

## 6 Conclusion

In this work, sensor deployment with target coverage is considered in order to maximize the network lifetime of the WSN. The proposed heuristic algorithm is capable of adjusting its sensing range and location by itself to monitor more number of targets. Performance evaluation shows that, the proposed heuristic algorithm has better network lifetime than the existing algorithms in both simple and k-coverage scenarios. Scheduling the wireless sensors, further improves the network lifetime.

## References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). Wireless sensor networks: A survey. *Journal of Network and Computer Applications*, *38*(4), 393–422.
2. Gungor, V. C., Lu, B., & Hancke, G. P. (2010). Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics*, *5*(10), 3557–3567.
3. Cardei, M., & Wu, J. (2006). Energy-efficient coverage problems in wireless ad hoc sensor networks. *Journal of Computer Communications and Sensor Networks*, *29*(4), 413–420.
4. Shih, K. P., Chen, H. C., Chou, C. M., & Liu, B. J. (2009). On target coverage in wireless heterogeneous sensor networks with multiple sensing units. *Journal of Network and Computer Applications*, *32*(4), 866–877.
5. Han, X., Cao, X. G., Loyd, E. L., & Shen, C. C. (2010). Fault-tolerant relay node placement in heterogeneous wireless sensor networks. *IEEE Transactions on Mobile Computing*, *9*(5), 643–656.
6. Krause, A., Rajagopal, R., Gupta, A., & Guestrin, C. (2011). Simultaneous optimization of sensor placements and balanced schedules. *IEEE Transactions on Automatic Control*, *56*(10), 2390–2405.
7. Yang, D., Misra, S., Fang, X., Xue, G., & Zhang, J. (2012). Two-tiered constrained relay node placement in wireless sensor networks: Computational complexity and efficient approximations. *IEEE Transactions on Mobile Computing*, *11*(8), 1399–1411.
8. He, X., Yanh, H., & Gui, X. (2010). The maximum coverage set calculated algorithm for WSN area coverage. *Journal of Networks*, *5*(6), 650–657.
9. Ammari, H. M., & Das, S. K. (2012). Centralized and clustered k-coverage protocols for wireless sensor networks. *IEEE Transactions on Computers*, *61*(1), 118–133.
10. Ashouri, M., Zali, Z., Mousavi, S. R., & Hashemi, M. R. (2012). New optimal solution to disjoint set K-coverage for lifetime extension in wireless sensor networks. *IET Wireless Sensor Systems*, *2*(1), 31–39.
11. Jia, S. G., Lu, L. P., Su, L. D., Xing, G. L., & Zhai, M. Y. (2013). An efficient sleeping scheduling for save energy consumption in wireless sensor networks. *Advanced Materials Research*, *756*(759), 2288–2293.
12. Chen, C. P., Mukhopadhyay, S. C., Chuang, C. L., Liu, M. Y., & Jiang, J. A. (2015). Efficient coverage and connectivity preservation with load balance for wireless sensor networks. *IEEE Sensors Journal*, *15*(1), 48–62.
13. Kulkarni, R. V., & Venayagamoorthy, G. K. (2011). Particle swarm optimization in wireless sensor networks: A brief survey. *IEEE Transactions on Systems, Man, and Cybernetics, System*, *41*(2), 262–267.
14. Mini, S., Udgata, S. K., & Sabat, S. L. (2014). Sensor deployment and scheduling for target coverage problem in wireless sensor networks. *IEEE Sensors Journal*, *14*(3), 636–644.
15. Bojkovic, Z., & Bakmaz, B. (2008). A survey on wireless sensor networks deployment. *WSEAS Transactions on Communications*, *7*(12), 1172–1181.
16. Hao, J., Zhang, B., Jiao, Z., & Hashemi, M. R. (2015). An adaptive compressive sensing based sample scheduling mechanism for wireless sensor networks. *Pervasive and Mobile Computing*, *2*, 113–125.
17. Silberschatz, A., Galvin, P. B., & Gagne, G. (1998). *Operating system concepts*. Boston: Addison-Wesley.
18. Yuan, X., & Duan, Z. (2009). Fair round-robin: A low complexity packet schduler with proportional and worst-case fairness. *IEEE Transactions on Computers*, *58*(3), 365–379.
19. Maleki, I., Khaze, S. R., Tabrizi, M. M., & Bagherinia, A. (2013). A new approach for area coverage problem in wireless sensor networks with hybrid particle swarm optimization and differential evolution algorithms. *International Journal of Mobile Network Communications & Telematics*, *3*(6), 61–67.
20. Zorbas, D., & Razafindralambo, T. (2013). Prolonging network lifetime under probabilistic target coverage in wireless mobile sensor networks. *IEEE Transactions on Computer Communications*, *36*(9), 1039–1053.
21. Zhou, F. (2011). Energy-efficient coverage using sensors with continuously adjustable sensing ranges. *Seventh international conference on natural computation (ICNC)*, 109–113.
22. Gu, y., Liu, H., & Zhao, B. (2007). Target coverage with QoS requirements in wireless sensor networks. *International conference on intelligent pervasive computing*, 35–38.
23. Medagliani, P., Leguay, J., Ferrari, G., Gay, V., & Lopez-Ramos, M. (2012). Energy-efficient mobile target detection in wireless sensor networks with random node deployment and partial coverage. *IEEE Transactions on Computer Communications*, *8*(3), 429–447.

24. Zhao, Q., & Gurusamy, M. (2008). Lifetime maximization for connected target coverage in wireless sensor networks. *IEEE/ACM Transactions on Networking*, *16*(6), 1378–1391.
25. Kim, H., & Han, S. (2015). An efficient sensor deployment scheme for large-scale wireless sensor networks. *IEEE Communications Letters*, *19*(1), 98–101.
26. Turkogullari, Y. B., Aras, N., Altinel, I. K., & Ersoy, C. (2010). A column generation based heuristic for sensor placement, activity scheduling and data routing in wireless sensor networks. *European Journal of Operational Research*, *207*(2), 1014–1026.
27. Hong, Y., Du, H., Li, D., & Chen, W. (2012). Conflict-aware data aggregation scheduling in wireless sensor networks with adjustable transmission range. *Discrete Mathematics, Algorithms and Applications*, *4*(3), 1–17.
28. Jia, J., Chen, J., Chang, G., & Tan, Z. (2009). Energy efficient coverage control in wireless sensor networks based on multi-objective genetic algorithm. *Computers & Mathematics with Applications*, *57*(11), 1756–1766.
29. Mao, X., Chen, H., Qiu, P., & Zhang, Z. (2010). Energy-efficient scheduling for multiple access in wireless sensor networks: A job scheduling method. *Computer Networks*, *54*(13), 2137–2146.
30. Karaboga, D., & Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied Soft Computing*, *8*, 687–697.
31. Sabat, S. L., Udgata, S. K., & Ajith, A. (2010). Artificial bee colony algorithm for small signal model parameter extraction of MESFET. *Engineering Applications of Artificial Intelligence*, *23*(2), 689–694.
32. Riley, J., Greggers, U., Smith, A., Reynolds, D. R., & Menzel, R. (2005). The flight paths of honeybees recruited by the waggle dance. *Nature*, *435*, 205–207.
33. Lee, J. W., Choi, B., & Lee, J. (2011). Energy efficient coverage of wireless sensor networks using ant colony optimization with three types of pheromones. *IEEE Transactions on Industrial Informatics*, *7*(3), 419–427.
34. Lin, Y., Zhang, J., Chung, H. S. H., Ip, W. H., Li, Y., & Shi, Y. H. (2012). An ant colony optimization approach for maximizing the lifetime of heterogeneous wireless sensor networks. *IEEE Transactions on Systems, Man, and Cybernetics, Applications and Reviews*, *42*(3), 408–420.
35. Liao, W. H., Kao, Y., & Wu, R. T. (2011). Ant colony optimization based sensor deployment protocol for wireless sensor networks. *Expert Systems with Applications*, *38*(6), 6599–6605.

**Krishnan Muralitharan** received the B.Sc., degree in Electronics and M.C.A., degree in Computer Science from Bharathiar University, Coimbatore, Tamilnadu, India, in 2001 and 2004, respectively. He was a Senior Software Engineer in Nilgiri Networks Pvt Ltd., [TeNet group (I.I.T Madras)], Tamilnadu, India from 2004 to 2009 and worked as a Senior Engineering Research Fellow (SERF) in the Department of Computer Applications, Anna university of Technology, Coimbatore, India from 2009 to 2012. Currently, he is working towards his Ph.D. degree in Department of Mathematics, Sri Ramakrishna Institute of Technology, Coimbatore, India. from July 2012 onwards. His current research interests includes smart grid networks, wireless sensor networks and optimization techniques.

**Vishnuvarthan Rajagopal** received his Bachelor's degree in Computer Science and Engineering from Bannari Amman Institute of Technology, Sathyamangalam, Tamilnadu, in 2013, and Master of Engineering from Anna University Regional Campus, Coimbatore in 2016. Currently, he is working as a Assistant Professor at United Institute of Technology, Coimbatore, Tamilnadu, India. His current research interests includes Wireless Sensor Networks and Content Centric Networking.

**Rathinasamy Sakthivel** is a professor in the Department of Mathematics, Sri Ramakrishna Institute of Technology, Coimbatore, India. He received the B.Sc., M.Sc., and Ph.D. degrees in mathematics from Bharathiar University, Coimbatore, India, in 1992, 1994, and 1999, respectively. Soon after the completion of his Ph.D. degree, he served as a lecturer at the Mathematics Department at the Sri Krishna College of Engineering and Technology, India. From 2001 to 2003, he was a post-doctoral fellow at the Mathematics Department, Inha University, South Korea. He was a visiting fellow at Max Planck Institute, Magdeburg, Germany, in 2002. From 2003 to 2005, he was a JSPS (Japan Society for the Promotion of Science) fellow at the Department of Systems Innovation and Informatics, Kyushu Institute of Technology, Japan. After that he worked as a research professor at the Mathematics Department, Yonsei University, South Korea, till 2006. Then he was a post-doctoral fellow (Brain Pool Program) at the Department of Mechanical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, Korea, from 2006 till 2008. In 2008, he joined Sungkyunkwan University, Suwon, as an Assistant Professor in mathematics and subsequently was promoted to an Associate professor of mathematics in 2012. He has published more than 190 research papers in reputed Science Citation Index journals. His major research areas include control theory, stability, robust control for uncertain systems and neural networks.