



Flower Pollination Algorithm with Powell's Method for the Minimum Energy Broadcast Problem in Wireless Sensor Network

M. Rajeswari¹ · KalaiPriyan Thirugnanasambandam² · R. S. Raghav³ · U. Prabu⁴ · Devaraj Saravanan⁵ · Dinesh Kumar Anguraj⁶

Accepted: 4 February 2021 / Published online: 25 February 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC part of Springer Nature 2021

Abstract

Minimum energy broadcast (MEB) problem in wireless sensor network has attracted attentions of the many researchers due to the limited bandwidth of the network and battery life of the sensor nodes. The data in a wireless network are transmitted from the source node to all other nodes and seek broadcast scheme to transmit with minimum energy consumption. The main objective of MEB is to minimize the transmission energy consumption of the network and is considered as an NP-complete problem. This work proposes a new variant of Flower pollination algorithm based on Powell's method (PFPA) to solve MEB problem in wireless sensor networks. The proposed algorithm is compared with other heuristic approaches and the performance of the algorithm is assessed using benchmark instances with 50 and 100 nodes. The effectiveness and merit of the proposed algorithm is demonstrated in terms of performance metrics.

Keywords Wireless sensor networks · Minimum energy broadcast · Flower pollination algorithm · Powell's method

1 Introduction

Wireless sensor network (WSN) is a cost effective ambience and many researchers had done work in the past decade. The information in WSN is broadcasted among all the nodes in the network and limited energy resources are the fundamental aspects. Modern technologies in WSN had made sensor nodes to communicate with outside world. The nodes in WSN are connected and shared under common transmission media to exchange data. The applications of WSN include manufacturing productivity, home security, and emergency outlet etc. The responsibility of the sensor nodes is to detect, gather data and transfer to the other network. The battery is the power source for the sensor nodes and some nodes are non-renewable or non-rechargeable. The most issue in a wireless network is maximizing the lifetime of the network and minimizes the coverage loss. The data are broadcasted to all

✉ Dinesh Kumar Anguraj
dineshinnov@outlook.com

Extended author information available on the last page of the article

other nodes in the transmission range of sensor node by the evident of wireless multicast advantage (WMA) [22]. Each node in the wireless network has omni-directional antenna and receive data within the transmission range. When a node is in transmission range, it can receive and transmit a signal through multi-hop communication. Multi-hop communication uses rely on nodes as intermediate links to transmit signals. When the transmission range of a node is high it can transmit data to a maximum number of nodes, but it consumes more energy of the wireless node.

In wireless networks, broadcasting involves node with omni-directional antenna under single transmission range. The transmission power of the sensor node is adjusted so that the transmission can reach to the farthest node in the network. So the transmission range will cover all the sensor nodes under WMA. The objective of MEB is to minimize energy consumption transmission range to transfer.

In this work, the author proposed a new variant of Flower pollination algorithm based on Powell's method proposed called Powell's Flower pollination algorithm (PFPA). Flower Pollination algorithm (FPA) is adapted to solve NP-hard combinatorial optimization problems by incorporating heuristic method. By embedding conjugate directions in Powell's method, the basic search pattern was improved towards direct search. In FPA there is inefficiency towards local search process and it result in poor exploitation process. To balance both exploration and exploitation process in FPA the authors incorporated heuristic technique in exploitation process of FPA. Powell's conjugate gradient descent method is used to find local minimum and it does bi-directional search towards search vector.

Yang [36] evaluated the performance of FPA with genetic algorithm and particle swarm optimization algorithm. The simulation results show that FPA outperforms both and it is successfully used in several fields [4]. However, for larger-scale problems, the performance of FPA degrades substantially due to the higher likelihood of trapping in locally optimal solutions. The proposed method alleviates this drawback by increasing the balance of diversification and intensification of the search process.

The remainder of the paper is organized as follows: Sect. 2 backgrounds the approaches solving MEB to attain optimal solution has discussed. In Sect. 3, the formulation of MEB is presented. In Sect. 4 the proposed flower pollination algorithm based on Powell's method is discussed. In Sect. 5 applicability of MEB has been explained with experimental result analysis. Finally, Sect. 6 concludes the research work.

2 Literature Review

Over the past decades, various researchers have attempted to solve MEB problems. This problem has been tracked by heuristic, local search, and meta-heuristics algorithms. The simple heuristics to find a minimum spanning tree are proposed based on prim's algorithm [32] and they used basic greedy search method. Other examples based on heuristic approaches are Greedy Perimeter Broadcast Efficiency (GPBE) [13], Broadcast Incremental Power [33], Adaptive Broadcast Consumption (ABC) [17] and Directed Bee colony algorithm is used to solve scheduling problem [27].

Wieselthier et al. proposed in the wireless network algorithms are based on heuristic methods and in wired network, the algorithms are based on link approaches. The author's proposed broadcast incremental power (BIP) algorithm with link-based approaches [33]. The BIP will iteratively develop arborescence for each iteration and add a new node to the tree thus results in minimum transmission power into partial

arborescence. The non-leaf nodes in the trees are arranged to ascend by the method of sweep procedure to reduce the transmission power to the farthest node. The sweep procedure should be implemented in multiple times and the results show little improvement.

Marks et al. developed an r-shrink procedure for solving MEB problem. In this procedure, the farthest node in the tree is removed one by one and assigned to some other transmission range nearer to so as to reduce the transmission power in the network. The total cost is calculated if the total cost is less than the original cost then arborescence is modified else it is retained. If the arborescence is modified, the whole process is repeated from the new arborescence [7].

Many researchers have also suggested using local search to obtain the near optimal solution like Broadcast Incremental Decremental Power (BIDP) [9]. Čagalj et al. proposed embedded wireless multicast advantage (EWMA). In this procedure, the transmission power of the node is increased to reduce the total cost of the transmission power to stop some other node to transmit entirely [5]. Kang and Poovendra used EWMA in largest expanding sweep search (LESS) procedure to reduce transmission power by increasing the transmission power of a particular node [14]. Raju et al. [28] used bio-inspired algorithm for solving cloud computing environment.

Kang and Poovendran used a perturbation based algorithm and LESS approach for local search to solve MEB in wireless ad-hoc networks [15]. Montemanni et al. proposed simulated annealing based r-shrink to solve MEB. The authors have used sweep procedure based simulated annealing to solve MEB in wireless networks [24]. Al-Shihabi et al. proposed a hybrid algorithm which combines r-shrink based partitioning with LP and integer programming. In this algorithm, nested partitioning part is implemented by generic partitioning [1].

The most recent trends in solving involves utilizing meta-heuristics due to their ability to cope with large-scale problems. Many works related to meta-heuristics algorithm utilizing the benefits of local search methods have done in different applications [12, 19, 23]. Das et al. proposed ant-colony based algorithm to solve MEB in wireless sensor network [6]. Wolf et al. proposed a hybrid algorithm based on evolutionary local search which uses r-shrink and mutation operator to randomly increase the transmission range of the nodes [34]. The results obtained are compared with other localization algorithms and nested partitioning [1] which demonstrated reduced computational time to obtain near optimal solution.

Wu et al. proposed a genetic algorithm which uses permutation encoding and decoding in the broadcast environment for minimum spanning tree problems [35]. They used rank scheme and different crossover operators and compared the result with simulated annealing [24]. Some of the works related to meta-heuristic algorithms on solving MEB includes Hybrid genetic algorithm [31] and Iterated local search algorithm [3]. Singh and Bhukya proposed a hybrid genetic algorithm which uses swap mutation and cycles crossover along with local search algorithm. The authors had used the r-shrink procedure for the local search operation and the combinations of r-shrink with genetic operators are used to achieve the optimal solution [31].

Guo and Yang provided the detailed survey on solving MEB problem with Omnidirectional and directional antennas for wireless ad-hoc networks [8]. Sengupta et al. described the objectives to deploy sensor nodes in the wireless network. Before deploying the sensor node have to consider, a number of sensor nodes should be minimized to minimize the cost of deployment. Maximize the coverage area and minimize the energy consumption of the sensor node. The lifetime of the network should be maximized [30].

Enan et al. solved routing problem in wireless sensor network and the authors proposed an evolutionary-based clustering algorithm and the transmission is based on single-hop communication. They formulated the network to achieve maximum stability period until parent node dies and minimized the energy consumption of the network [16].

To minimize energy broadcast problem, the above discussed approaches endured to obtain the optimal solution. Computational time is most needed to validate the performance of the algorithm over the various sized network. The objective of this research is to obtain an optimal solution within an acceptable computational time.

3 Problem Formulation

In wireless networks, broadcasting allows all the sensor nodes to share the data proficiently with other nodes in the transmission medium. To construct broadcast trees, there is a limited energy resource because the nodes are operated on the battery. The network consists of a set of nodes, in which a node is considered as a source node, the objective of the minimum energy broadcast (MEB) problem is to minimize the total energy consumption of the network and it has proved to be an NP-Complete problem [5, 20]. In this network, all the nodes have the capacity to regulate their transmission range. The transmission range is assigned to each node in the network and every other node inside the transmission range will receive the message. The goal of the MEB is to assign transmission ranges to the nodes such that the total energy consumption is minimized.

The MEB problem in a wireless network can be stated using directed complete graph $G = (V, E)$, where V represents the set of nodes in the wireless network and E represents set of edges connecting vertices. The energy required to set up a link from node i to node j is denoted as φ_{ij} where the node links are the edges of the graph $ij \in E$. In wireless network the broadcasting done using Omni-directional antenna and nodes are covered in single transmission range. For example if node p is transmitting data to the node q then each other node i in the transmission range will also receive the same data sent by the node p such that $\varphi_{pi} \leq \varphi_{qj}$.

The objective of the MEB problem is to find the minimum transmission energy broadcast routing tree, source node $s \in V$ broadcast message either directly or through intermediate nodes to all other remaining nodes of V . The start node of arborescence is responsible to relay the broadcast message to all other terminal nodes of the same directed edge. The transmission energy required for node p in arborescence is the power of the node to transmit to the farthest child node p (longest edge) in the tree. The transmission energy required for any leaf node is considered to be zero, since they are not relaying any message to other nodes in the tree.

The energy required to send the packets can be measured using

$$\varphi = \zeta d^\psi \quad (1)$$

To broadcast, the transmission energy required by the node in arborescence can be calculated by adding transmission energy of each parent node in the tree. In a tree, with source node s , the minimum total energy required can be measured by using Eq. (2)

$$TE_{sol_i} = \sum_{a \in V} \max_{(a,b) \in E_{sol_i}} d(a,b)^\psi \quad (2)$$

The solution consists of minimum energy broadcast trees which are generated by the algorithm after each iteration. The transmission energy generated by each solution is represented as TE_{sol_i} and i is the index for the solution number. The path loss exponent constant ψ lies between 2 and 4 [18] and power threshold ζ is normalized to 1. $d(a, b)$ is the distance between the nodes a and b . The distance $d(a, b)$ in a transmission, the received power signal at b varies and can be calculated using Euclidean distance by

$$d(a, b)^\psi = \left(\left[(p_a - p_b)^2 + (q_a - q_b)^2 \right]^{\frac{1}{2}} \right)^\psi \quad (3)$$

The Euclidean distance (p_a, p_b) between the nodes a and b are the coordinates of the node p and q . The minimum transmission energy contains the near optimal solution.

4 Flower Pollination Algorithm with Powell's Method

4.1 Characteristics of Flower Pollination

All flowering plants have a peculiar type of reproduction namely pollination. The exchanges of pollens between the flowers are through insects, birds, etc. Pollination process is made of two types namely biotic and abiotic. Among these types, 0.9 ratio of pollination is done through biotic structure in which insects and birds transfers the pollens to other flowers. A tenth ratio of pollination is done through abiotic pollination, which is the process carried through wind and dissemination offer assistance. Pollen vectors are the pollinator which works in terms of diversity.

In the cross-pollination, biotic occurs at longer distance with the pollinators like bees, birds, insects, and other animals can travel for long geographical region. These types of pollination can be represented as global pollination. A randomized walk is done using Levy flight [25] which also imposes a diversified search in the given region. Objective in flower pollination is to reproduce the production using pollens and to retrieve the best pollinated flowers based on their attributes. Survival of the fittest and providing optimal reproduction in terms of number and fittest. This is the optimization process to achieve optimal reproduction of the flowering plants.

4.2 Flower Pollination Algorithm

Xin-She Yang proposed flower pollination algorithm (FPA) [36], inspired by the natural pollination process of flowering plants.

Rules for the simplicity of FPA:

1. Global pollination process: This process takes place through biotic and cross-pollination, the movement of pollen carrying pollinators obeys Levy flights behavior.
2. Local pollination process: This process takes place through abiotic and self-pollination.
3. Local neighborhood: The pollinators are comparable to offspring probability proportional to the similarity of the two flowers.
4. Switch probability controls the interaction of local and global pollination and lies between 0 and 1. Here the higher probability is indicated for plants in the neighborhood which mimics local pollination.

In nature, each plant can have many number of flowers and each flower patch will release billions of pollen gametes. For simplicity, the authors assume every plant has single flower and it generate single pollen gamete. Thus there is no need to differentiate a flower, pollen, a plant or solution to the problem. The solution \mathcal{F}_i is equal to the pollen or flower. In FPA, there are two steps they are global and local pollination.

4.2.1 Global Pollination

In the global pollination, the pollen gametes from the flowers are carried by the pollinators such as birds, bats, and bees are traveled to the longer distance. This process ensures reproduction and pollination of the fittest Ψ_b .

Therefore the Rule 1 and flower constancy in Rule 3 can be mathematically represented as

$$\mathcal{F}_i^{t+1} = \mathcal{F}_i^t + \varphi Lev(\alpha)(\Psi_b - \mathcal{F}_i^t) \tag{4}$$

where \mathcal{F}_i^t is the solution vector \mathcal{F}_i at iteration t for the pollen i , and Ψ_b is the best solution at current iteration or generation. Here φ is the scaling factor to control step size and $Lev(\alpha)$ is the parameter, strength of the pollination which is Levy-flights based step size. Since the insects travel various distance in global pollination, a Levy flight can simulate this behaviour [29]. Xin-she Yang used levy distribution when $Lev > 0$

$$Lev(\alpha) \sim \frac{\alpha \Gamma(\alpha) \sin\left(\frac{\pi\alpha}{2}\right)}{\pi} \frac{1}{s^{1+\alpha}}, \quad \text{where } s \gg s_0 \gg 0 \tag{5}$$

where $\Gamma(\alpha)$ is gamma function and with larger step where $s > 0$ and the value of s_0 is as small as 0.1. To generate a pseudo-random step size which obeys the Levy distribution is not trivial. The random numbers for step size s are generated by Mantegna algorithm using Gaussian distributions G and H can be represented as

$$s = \frac{G}{|H|^{1/\alpha}}, \quad G \sim N(0, \sigma^2), \quad H \sim N(0, 1) \tag{6}$$

where $G \sim N(0, \sigma^2)$ represents Gaussian normal distribution with a variance of σ^2 and zero mean. The variance for the function can be calculated as

$$\sigma^2 = \left[\frac{\Gamma(1 + \alpha)}{\alpha \Gamma((1 + \alpha)/2)} \cdot \frac{\sin\left(\frac{\pi\alpha}{2}\right)}{2^{(\alpha-1)/2}} \right]^{\frac{1}{\alpha}} \tag{7}$$

Mantegna algorithm is used to produce a random sample which obeys distribution mathematically [21].

4.2.2 Local Pollination

The local pollination process both Rule 2 and Rule 3 can be mathematically represented using Eq. (8)

$$\mathcal{F}_i^{t+1} = \mathcal{F}_i^t + \eta(\mathcal{F}_k^t - \mathcal{F}_l^t) \tag{8}$$

where \mathcal{F}_k^t and \mathcal{F}_l^t are the pollen of neighboring flowers of the same plant, that is it is extracted from the same population with less flower constancy in neighborhood. Mathematically, when \mathcal{F}_k^t and \mathcal{F}_l^t comes from the same plant or from same population, then it performs a local random walk η using uniform distribution of [0, 1]. The flower pollination pursuits occur at global and local scales. In nature, the adjacent flower patches or flower in the not-so-far-away neighborhood transfer the pollen grains by local flower pollen than global. To mimic this feature, we switch between local and global pollination. The switch probability $Pe[0,1]$ Rule 4 or proximity probability is used intensively. The probability value $P = 0.5$ is used as an initial value and $P = 0.8$ may work better in many applications.

4.3 Powell's Method

The Powell's method is the extension of the basic pattern search method for direct search and can be improved by the method of conjugate directions. In Powell's method for two variable functions, the function is first minimized towards the direction of coordinates and start with second coordinate thus Powell's pattern is formed.

There is inefficiency in FPA, which is good at exploration but poor at exploitation process. The local search in FPA increases with increase in a number of iteration. To address this concerning problem, the local pollination phase is done using Powell's method [26]. M.J.D. Powell proposed Powell's conjugate gradient descent method to find a local minimum of a function by bi-directional search towards search vector. The new candidate position is in search vector and the displacement vector is added at the end of search list. The best search vector is deleted from the search vector list which contributed mainly towards a new direction.

The conjugate gradient descent method directs towards the nearly given point of search. This proposes strong local search capability and it is incorporated into FPA algorithm to improve the process of local search and increase the convergence speed and accuracy. In every iteration, the population position after updating is not directly fed to the next iteration process. Instead, conjugate gradient descent directions is employed at \mathcal{F}_i by the conjugate factor β_i . The population position \mathcal{F}_i of the flower decreases sufficiently within a predetermined number of iterations. The proposed algorithm combines Powell's method and FPA algorithm to deliver strong local and global searching capabilities.

Let

$$Y = Y(\mathcal{F}_1, \dots, \mathcal{F}_n) \tag{9}$$

$$C = \nabla Y = \frac{\partial Y}{\partial \mathcal{F}_1} i_1 + \frac{\partial Y}{\partial \mathcal{F}_2} i_2 + \dots + \frac{\partial Y}{\partial \mathcal{F}_n} i_n \tag{10}$$

$$d_i = -C_i + \beta_i d_{i-1} \tag{11}$$

where d_i is the deflection parameter, the next iteration is employed by multiplying conjugate factor β_i at gradient and adding to the negative gradient.

$$\beta_i = \left[\frac{\|C_i\|}{\|C_{i-1}\|} \right]^2 \tag{12}$$

$$\|C\| = \sqrt[2]{\frac{\partial Y^2}{\partial \mathcal{F}_1} + \dots + \frac{\partial Y^2}{\partial \mathcal{F}_n}} \tag{13}$$

Evaluate τ ,

$$\mathcal{F}^0 = (\mathcal{F}_1, \dots, \mathcal{F}_n)^0 \tag{14}$$

$$\mathcal{F}^1 = \mathcal{F}^0 + \tau d_0 \tag{15}$$

Update X^{i+1}

$$\mathcal{F}^{i+1} = \mathcal{F}^i + \tau d_i \tag{16}$$

where C_i the objective function and this method has the characteristic of the second termination and reaches the minimum point after limited generations or iterations. The pseudo code of Powell’s method and proposed PFPA is shown in Algorithms 1 and 2. The workflow of Powell’s method and modified PFPA to solve MEB in wireless sensor network are shown in Figs. 1 and 2.

Algorithm 1 Algorithm for Powell’s method.

Algorithm 1: Powell’s Method

Initialize the starting point \mathcal{F}_1 , independent vectors $d_i = e_i (i = 1, 2, \dots, V)$, set $f(1) = f(\mathcal{F}_1), \mathcal{F}_c(1) = \mathcal{F}_1, m = 1$

While (stopping criteria met) **do**

For $i=1:V$ **do**

If ($m \geq 2$) **then**

$d_i = d_{i+1}$

End if

$\mathcal{F}_{i+1} = \mathcal{F}_i + \mu_i d_i$

where μ_i is determined by minimizing $f(\mathcal{F}_{i+1})$,

End for

$d_{i+1} = \sum_1^V \mu_i d_i = \mathcal{F}_{V+1} - \mathcal{F}_V$

$\mathcal{F}_c(m+1) = \mathcal{F}_{V+1} + \mu_m d_{i+1}$

$f(m+1) = f(\mathcal{F}_c(m+1))$

$m = m + 1, \mathcal{F}_1 = \mathcal{F}_c(m)$,

$\Delta f = f(m) - f(m-1)$

End While

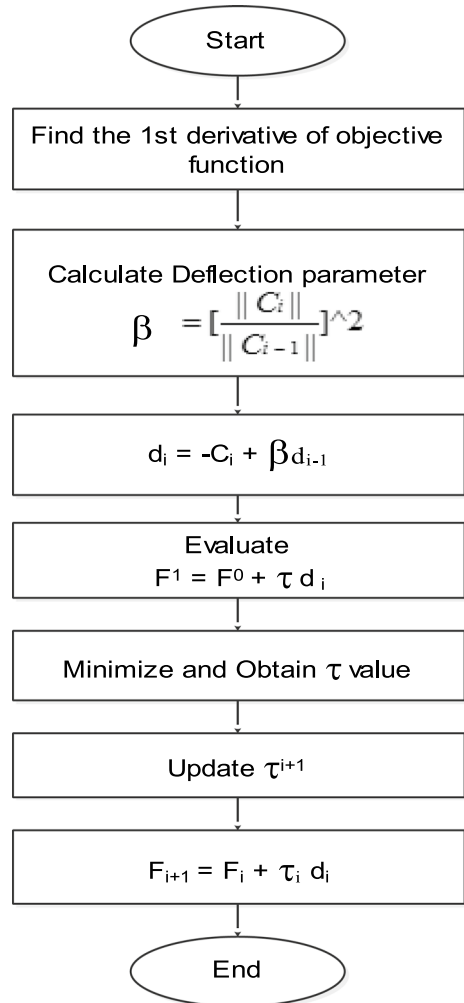
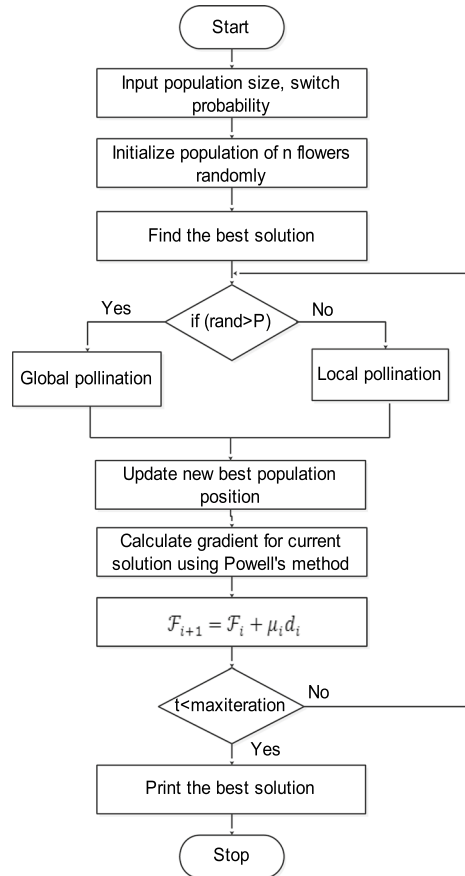
Fig. 1 Workflow of Powell's method

Fig. 2 Workflow of PFPA method



Algorithm 2 Algorithm for PFFA.

Algorithm 2: Powell's Flower Pollination Algorithm

Objective function $f(\mathcal{F})$, $\mathcal{F} = (\mathcal{F}_1, \mathcal{F}_1, \dots, \mathcal{F}_d)$
 Initialize the population of n flowers randomly
 Find the best solution Ψ_b
 switch probability $Pe[0,1]$
While ($t < maxgeneration$) **do**
 For $i = 1:n$ **do**
 If $rand < P$ **then**
 Global pollination using Levy
 Distribution
 $\mathcal{F}_i^{t+1} = \mathcal{F}_i^t + \varphi Lev(\alpha)(\Psi_b - \mathcal{F}_i^t)$
 Else
 Draw η using Uniform distribution $[0,1]$
 Local pollination
 $\mathcal{F}_i^{t+1} = \mathcal{F}_i^t + \eta(\mathcal{F}_k^t - \mathcal{F}_i^t)$
 End if
 Evaluate \mathcal{F}_i^t , replace \mathcal{F}_i^t by \mathcal{F}_i^{t+1} update
 new solution in the population
 End for
 Update the current best solution \mathcal{F}_i^t
 Calculate the gradient for current solution
 using Powell's Method (Algorithm 1)
End while
 Print the best solution

5 Experimental Analysis

5.1 Environment Setup

The performance of the proposed PFFA for solving MEB is evaluated using instances taken from DAG [1] available from <http://dag.informatik.uni-kl.de/research/meb/>. There are 30 instances of each size with 20, 50 and 100 numbers of nodes, which are randomly scattered in the 1000×1000 square. The experimentation to solve MEB was done on different optimization algorithm under similar environment conditions to measure the performance of the proposed algorithm. The algorithm to solve MEB is coded using MATLAB 2012 platform on Windows Intel c2 GHz Core 2 quad processor with 2 GB of RAM. The simulation is done with the parameter setting as shown in Table 1 and appropriate parameter values are assigned based on preliminary experiments. To evaluate the efficiency of the proposed algorithm on solving MEB, competitor algorithms chosen are particle swarm optimization algorithm (PSO) [11], ant colony optimization (ACO) [10], hybrid genetic algorithm (HGA) [31], memetic algorithm (MA) [2], iterated local search (ILS) [15], evolutionary local search (ELS) [34].

The simulation result for the 10-node network is not given in this work; the proposed algorithm PFFA achieved optimum results for all the nodes. The simulation results for solving MEB using PFFA on varying number of node size are represented in Table. The

Table 1 Parameter setting for experimental evaluation

Type	Method
Population size	100
Maximum iterations	500
Run	20
Termination condition	Maximum iteration
Heuristic	Powell's method
Path loss ψ	2
Power threshold ζ	1

nodes are randomly distributed and the simulation is done for 20 runs. There is no deviation in results for all executions and the deviation is represented in excess percentage.

5.2 Performance Metrics

The performance of the proposed method PFFA is assessed by comparing with other algorithms. Here, in this work, the authors have considered four performance metrics to calculate the significance of the proposed system.

5.2.1 Found

Found specifies the number of times the algorithm is computed to obtain the best value out of 20 runs. To compute the best solution Euclidian distance is used. The best solutions are considered as the near to obtain best energy consumption value.

5.2.2 Excess (%)

The excess percentage will provide the deviation of solution from the optimal solution. The excess percentage for the solution can be represented by

$$Excess(\%) = \frac{1}{run} \sum_{i=1}^{run} \left[\frac{Best_value_i}{Optimal\ value} - 1 \right] * 100 \quad (17)$$

The excess (%) can be calculated for *run* number of times and *Best_value_i* is the best solution obtained at *i*th iteration and *optimal value* is the optimum value given in the dataset.

5.2.3 Computational Time

The computational time is the total time taken to complete the runtime of the proposed algorithm. The evaluation is based on random location and best solution is not epoch value for termination.

$$\text{Computational time} = \text{Runtime of proposed algorithm} \quad (18)$$

5.2.4 Standard Deviation

The standard deviation (SD) is the measure of dispersion of solution from its mean value. The standard deviation can be calculated by

$$SD = \sqrt{\sum_{i=1}^r (\text{value in each instance}_i - \text{mean value})^2} \quad (19)$$

The simulation result of PFFA on solving MEB is shown in Table 2. For a 50-node network, out of 30 instances, our proposed algorithm have found best results for 24 instances with 0.06% of excess %, which mean 0.06% of instances deviate from the optimal result. The computation time taken for 50-node instance to obtain an optimal result is 1.53 s. For a 100-node network, our proposed algorithm found best values for 17 instances out of 30 instances with a computational time of 46.54 s. The deviation of solution from the optimal result is given by 0.25%.

The instances for a 50-node network are taken and compared with competitor algorithms on solving MEB. Table 3 reports the detailed performance analysis of proposed PFFA algorithm with other competitor algorithm. The algorithm run for 20 times and excess (%), found, computational times are calculated for all the instances. The performance of the proposed algorithm PFFA on solving MEB is validated by computational time and excess percentage. All the columns in the Table represents performance metrics and ‘-’ shows that the algorithm had obtained best value for the corresponding instances. The comparison is made for seven algorithms; ILO performs worst in terms of excess percentage and computational time. ELS performs worst in deviating the result and better with computational time. ACO performs better than MA and HGA with respect to both computational time and excess percentage. Still, there is a gap between the proposed algorithm and existing algorithm. PSO achieved a better result than other existing algorithm but when compared with proposed PFFA achieved six better results with marginal computational time and lesser excess percentage.

The simulation results and the performance evaluation of proposed algorithm with other existing algorithm on solving MEB for 100-node network are tabulated in Table 4. From the previous simulation result for 50 runs, only ACO, HGA and PSO gave a marginal result with proposed PFFA. So for 100-node network, the authors have considered ACO, HGA, and PSO to measure the performance of PFFA. Among these algorithms, HGA shows poor performance in terms of computational time and excess percentage. ACO achieved results in lesser computational time but the excess percentage is more compared to other algorithms. The proposed algorithm PFFA shows five best results than PSO algorithm with better computational time.

Table 2 Simulation results of PFFA

S. no	No. of nodes	Excess (%)	Time	Found
1	50	0.06	1.53	24 instances out of 30
2	100	0.25	46.54	17 instances out of 30

Table 3 Performance analysis of PPPA with other algorithms solving MEB for 50 modes

Instance	Optimum	ELS		HGA		ACO		ILO					
		E (%)	Found	T (s)	E (%)	Found	T (s)	E (%)	Found	T (s)			
p50.00	399,074.64	6.32	0/30	5.74	0.72	15/30	14.26	0.00	27/30	1.59	4.30	0/30	61.19
p50.01	373,565.15	1.33	0/30	4.42	0.59	16/30	7.53	0.61	20/30	4.22	3.98	0/30	98.64
p50.02	393,641.09	3.33	2/30	6.37	-	30/30	7.17	-	30/30	3.23	12.83	0/30	23.71
p50.03	316,801.09	4.77	0/30	4.98	-	30/30	10.78	-	30/30	2.00	7.19	1/30	10.28
p50.04	325,774.22	4.33	4/30	5.75	0.47	13/30	27.40	0.21	21/30	3.55	7.32	0/30	64.01
p50.05	382,235.90	3.00	4/30	3.84	-	30/30	23.10	-	30/30	0.53	3.20	0/30	44.85
p50.06	384,438.46	1.94	6/30	8.13	-	30/30	24.85	-	30/30	2.25	1.00	15/30	51.90
p50.07	401,836.85	6.44	2/30	6.10	0.93	13/30	23.17	0.03	27/30	1.75	7.90	0/30	79.47
p50.08	334,418.45	0.17	21/30	5.73	-	30/30	4.54	-	30/30	3.16	-	30/30	98.96
p50.09	346,732.05	3.05	0/30	3.20	0.24	13/30	25.12	-	30/30	4.00	11.95	1/30	37.13
p50.10	416,783.45	2.09	3/30	1.48	-	30/30	9.82	-	30/30	0.57	2.35	0/30	28.21
p50.11	369,869.41	3.74	1/30	7.68	0.29	21/30	26.18	0.57	28/30	1.38	4.92	0/30	88.27
p50.12	392,326.01	2.17	1/30	7.11	0.73	17/30	13.31	0.44	27/30	1.56	0.73	4/30	100.2
p50.13	400,563.83	2.03	1/30	5.46	0.12	19/30	11.67	0.00	25/30	3.05	1.34	1/30	53.35
p50.14	388,714.91	1.21	1/30	3.13	-	30/30	12.67	0.12	24/30	4.26	-	30/30	14.86
p50.15	371,694.65	0.59	0/30	5.52	-	30/30	23.47	-	30/30	2.49	1.64	0/30	16.71
p50.16	414,587.42	2.63	6/30	1.35	0.43	21/30	9.12	-	30/30	4.15	6.26	0/30	93.40
p50.17	355,937.07	6.31	0/30	2.63	0.38	28/30	11.61	-	30/30	1.70	5.53	0/30	95.22
p50.18	376,617.33	1.64	0/30	3.15	0.82	24/30	13.46	0.96	20/30	1.46	1.02	1/30	32.80
p50.19	335,059.72	8.72	9/30	7.60	-	30/30	29.12	-	30/30	1.19	10.70	0/30	16.39
p50.20	414,768.96	0.82	0/30	7.86	0.70	19/30	28.43	0.00	28/30	0.86	7.21	0/30	83.69
p50.21	361,354.27	2.26	23/30	7.15	-	30/30	25.67	-	30/30	4.24	-	30/30	9.80
p50.22	329,043.51	0.31	25/30	4.07	-	30/30	18.90	-	30/30	3.32	5.15	0/30	16.51
p50.23	383,321.04	2.73	11/30	7.47	-	30/30	11.66	0.99	27/30	3.56	2.62	0/30	88.58
p50.24	404,855.92	0.86	5/30	8.37	-	30/30	4.14	-	30/30	2.16	1.08	5/30	76.70

Table 3 (continued)

Instance	Optimum	ELS		HGA		ACO		ILO					
		E (%)	Found	T (s)	E (%)	Found	T (s)	E (%)	Found	T (s)			
p50.25	363,200.32	-	30/30	2.31	-	30/30	26.21	0.51	24/30	1.15	-	30/30	44.77
p50.26	406,631.51	7.39	0/30	3.62	0.91	21/30	16.97	0.06	22/30	4.24	12.36	0/30	56.24
p50.27	451,059.62	4.42	0/30	4.86	0.53	22/30	8.44	-	30/30	0.74	4.20	0/30	32.80
p50.28	415,832.44	2.09	5/30	6.82	0.55	25/30	14.60	0.54	21/30	3.90	0.80	1/30	60.62
p50.29	380,492.77	0.74	15/30	7.87	-	30/30	5.15	0.17	19/30	4.13	1.59	1/30	26.04
Average		3.02	5.83/30	5.33	0.56	24.56/30	16.28	0.35	27/30	2.55	4.97	4.8/30	53.51
Instance	MA	PSO		PPPA									
		E (%)	Found	T (s)	E (%)	Found	T (s)						
p50.00	1.55	10/30	28.00	0.50	25/30	0.90	0.22	26/30	1.02				
p50.01	0.99	14/30	18.84	-	30/30	1.23	-	30/30	0.99				
p50.02	0.62	12/30	27.90	-	30/30	1.77	-	30/30	1.18				
p50.03	-	30/30	5.98	-	30/30	2.50	-	30/30	1.25				
p50.04	40.75	8/30	4.81	0.95	21/30	1.34	0.05	29/30	0.83				
p50.05	0.90	14/30	24.23	-	30/30	1.09	-	30/30	1.26				
p50.06	-	30/30	6.37	-	30/30	1.62	-	30/30	2.09				
p50.07	1.65	11/30	11.49	0.20	26/30	2.44	-	30/30	1.92				
p50.08	-	30/30	10.08	-	30/30	1.05	-	30/30	1.01				
p50.09	0.90	14/30	7.90	-	30/30	2.20	-	30/30	1.03				
p50.10	-	30/30	25.52	-	30/30	1.80	-	30/30	1.36				
p50.11	0.39	23/30	14.76	0.53	28/30	2.99	-	30/30	1.44				
p50.12	1.73	20/30	11.55	-	30/30	1.94	-	30/30	1.33				
p50.13	0.87	11/30	12.26	-	30/30	2.83	-	30/30	2.44				
p50.14	0.23	21/30	7.23	0.05	28/30	2.54	0.02	29/30	1.31				

Table 3 (continued)

Instance	MA		PSO		PPPA		
	E (%)	T (s)	Found	T (s)	E (%)	Found	T (s)
p50.15	–	13.37	30/30	–	–	30/30	1.36
p50.16	1.79	14.32	23/30	0.75	0.02	26/30	1.74
p50.17	0.79	26.39	28/30	–	–	30/30	2.98
p50.18	0.06	25.17	21/30	0.04	–	30/30	1.52
p50.19	–	4.44	30/30	–	–	30/30	1.46
p50.20	0.61	14.11	7/30	0.55	0.02	28/30	2.79
p50.21	0.39	11.15	19/30	–	–	30/30	0.89
p50.22	0.48	10.62	13/30	–	–	30/30	1.57
p50.23	0.39	24.06	11/30	0.06	–	30/30	1.70
p50.24	0.26	15.60	10/30	0.78	–	30/30	1.94
p50.25	–	25.33	30/30	–	–	30/30	1.88
p50.26	1.07	29.41	24/30	–	–	30/30	0.98
p50.27	0.43	20.83	22/30	0.03	–	30/30	1.43
p50.28	0.32	29.52	21/30	–	–	30/30	0.89
p50.29	–	16.80	30/30	0.17	0.01	29/30	2.35
Average	2.60	16.60	19.9/30	0.38	0.06	29.56/30	1.53

E—represents the Excess % and T— represents the computational time in seconds, '–' shows that the algorithm attained optimal solution for the corresponding instance

Table 4 Comparison of PFFPA and other algorithms solving MEB for 100 nodes

Instance	Optimum	ACO			HGA			PSO			PFFPA		
		E (%)	SD	T (s)	E (%)	SD	T (s)	E (%)	SD	T (s)	E (%)	SD	T (s)
p100.00	340,869.27	0.63	123.17	51.17	0.64	16.18	64.93	-	0.00	51.92	-	0.00	63.26
p100.01	355,284.77	0.18	257.52	25.03	0.58	176.36	2.22	0.34	130.59	60.53	-	0.00	70.91
p100.02	377,145.59	-	-	50.33	0.69	61.45	5.09	-	0.00	50.35	-	0.00	60.98
p100.03	356,942.53	0.98	187.89	28.44	0.74	165.89	92.80	0.08	150.10	22.75	0.15	148.98	33.12
p100.04	384,446.36	0.66	170.49	15.27	1.09	33.94	68.38	-	0.00	28.50	-	0.00	27.55
p100.05	416,758.58	-	-	24.72	1.08	48.57	53.65	-	0.00	35.85	-	0.00	35.47
p100.06	376,408.49	0.80	1604.72	15.49	0.40	19.03	27.57	-	0.00	30.13	-	0.00	29.44
p100.07	343,798.46	-	-	54.14	0.20	1542.91	10.10	0.79	1456.64	36.26	0.14	1468.23	35.35
p100.08	372,254.06	0.88	744.53	8.04	1.00	499.31	45.54	0.38	459.61	48.80	0.63	462.26	59.16
p100.09	366,993.89	-	-	23.18	0.93	94.73	24.34	-	0.00	39.91	-	0.00	50.54
p100.10	334,579.00	-	-	31.83	0.41	69.77	17.10	-	0.00	61.81	-	0.00	72.01
p100.11	356,219.14	-	-	53.53	0.38	72.07	36.32	-	0.00	59.30	-	0.00	69.14
p100.12	393,854.17	0.01	620.11	10.70	0.49	441.31	61.16	0.47	358.19	15.76	0.14	429.93	25.97
p100.13	331,270.37	-	-	39.25	0.42	794.89	95.54	0.87	778.31	41.76	0.22	780.97	52.30
p100.14	344,175.57	0.12	1014.47	15.25	0.18	47.00	17.97	-	0.00	12.25	-	0.00	11.38
p100.15	352,884.55	0.86	76.83	48.62	0.81	103.67	59.60	0.14	34.58	60.65	0.11	31.88	60.09
p100.16	338,713.69	0.94	1008.50	9.14	0.86	569.41	78.23	0.89	541.17	17.30	0.29	538.83	16.54
p100.17	374,059.25	0.26	3363.92	49.56	0.78	1682.92	52.44	0.92	1611.64	55.91	0.18	1603.04	55.35
p100.18	331,926.13	0.63	2625.35	21.33	1.02	38.81	56.88	-	0.00	45.39	-	0.00	44.98
p100.19	365,078.37	-	-	38.83	0.84	65.94	32.14	-	0.00	41.48	-	0.00	40.82
p100.20	355,078.27	0.05	1538.95	38.64	0.26	2003.71	92.68	0.67	1929.15	57.87	0.22	1940.54	57.79
p100.21	362,204.29	0.58	181.42	18.67	0.17	289.75	99.33	0.89	272.51	60.37	-	0.00	60.03
p100.22	366,125.96	-	-	42.74	0.96	94.31	89.64	0.69	90.72	53.40	0.21	94.21	52.85
p100.23	409,062.55	0.05	1145.60	46.34	0.79	342.49	64.13	0.19	309.81	40.61	-	0.00	63.97
p100.24	357,772.11	0.83	829.06	8.22	0.20	510.11	76.38	0.24	418.89	26.16	-	0.00	50.79

Table 4 (continued)

Instance	Optimum	ACO		HGA		PSO		PFPA					
		E (%)	SD	T (s)	E (%)	SD	T (s)	E (%)	SD	T (s)			
p100.25	357,191.63	0.23	709.58	51.05	0.89	423.22	74.39	0.47	357.10	42.56	0.41	417.25	36.77
p100.26	352,148.02	-	-	44.93	0.72	707.66	32.87	0.42	668.23	51.38	-	0.00	53.34
p100.27	370,033.07	0.62	992.31	31.63	0.86	810.22	6.39	0.88	790.36	35.24	0.37	802.71	34.57
p100.28	348,889.36	0.62	1085.86	39.25	0.69	85.90	86.87	-	0.00	33.98	-	0.00	33.17
p100.29	357,595.04	0.71	421.17	33.08	0.72	497.53	62.70	0.90	487.96	38.76	0.27	479.41	38.64
Average	-	0.53	935.07	32.28	0.66	410.30	52.91	0.57	361.52	41.90	0.25	317.03	46.54

E- represents the Excess %, SD represents Standard deviation and T- represents the computational time in seconds, '-' shows that the algorithm attained optimal solution for the corresponding instance

Table 5 Comparison of PFPA with Competitor Algorithms

Algorithms	50 nodes			Algorithms	100 nodes		
	Excess (%)	Found	Time		Excess (%)	SD	Time
PFPA	0.06	29.56/30	1.53	PFPA	0.25	317.03	46.54
PSO	38	28.4/30	2.07	PSO	0.57	361.52	41.9
ACO	0.35	27/30	2.55	ACO	0.53	935.07	32.28
HGA	0.56	24.56/30	16.28	HGA	0.66	410	52.91
MA	2.6	19.9/30	16.6				
ELS	3.02	5.83/30	5.33				
ILO	4.97	4.8/30	53.51				

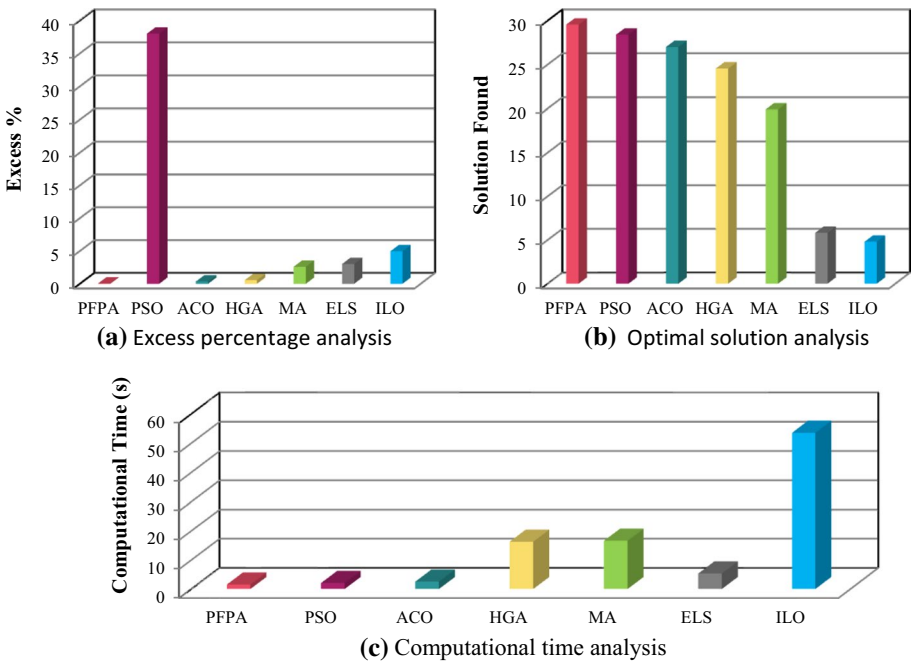


Fig. 3 The performance analysis for 50 node network

The overall performance analysis of the proposed algorithm PFPA with other algorithms on solving MEB over 50 and 100-node networks is shown in Table 5.

Figure 3 shows the comparison of performance analysis for proposed algorithm with existing algorithm. For a 50-node network, Fig. 3a shows the average excess % of PFPA compared with other algorithm. Figure 3b shows the average number of times the best solution is found by the proposed algorithm and other algorithm. Figure 3c shows the average computation time availed by the proposed algorithm PFPA for 20 independent runs. Inspecting the results in Fig. 3, it is evident that PFPA produce near optimal solutions in reduced generations for medium sized network.

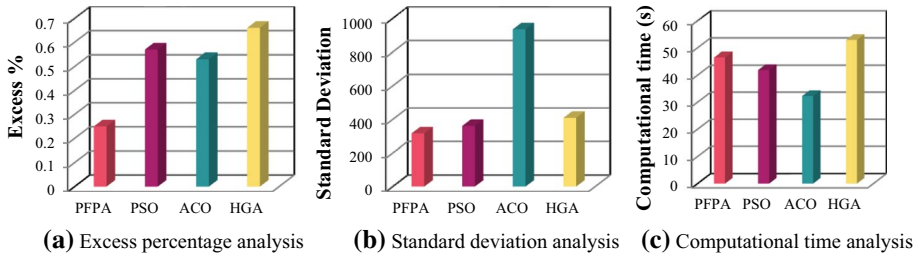


Fig. 4 The performance analysis for 100 node network

For 100 node network the comparison of performance analysis for proposed algorithm with existing algorithm is shown in Fig. 4. Figure 4a shows the average excess % of PFPA compared with competitor algorithms. Figure 4b shows the average standard deviation of the proposed algorithm and other algorithm. Figure 4c shows the average computation time consumed by the proposed algorithm PFPA for 20 independent runs. From the Figs. 3 and 4 it is evident that PFPA produce near optimal solutions in reduced generations for both medium and large sized network.

The research for solving NP-hard combinatorial optimization problem of MEB is carried out by the proposed algorithm PFPA. Various meta-heuristic algorithms are chosen for solving MEB and the results obtained are compared with proposed algorithm PFPA. The obtained results are compared with competitor methods and tabulated in Tables 3, 4 and 5. Investigating from Figs. 3 and 4, it is evident our proposed PFPA shows best value on performance metrics for a 50-node and a 100-node network compared with other competitor algorithms. Comparing with other metaheuristic algorithms, for a 50-nodes the excess percentage was reduced to 0.06%, the computation time to obtain best results was 1.53 ms and out of 30 solutions 24 solutions obtained best results. For a 100-node instance, PFPA excess percentage was reduced to 0.25% the computation time taken was 46.54 ms and out of 30 instances 17 instances found best solution.

The proposed PFPA algorithm is tested on various sized datasets and from the result it is proved PFPA outperforms well than other competitor algorithms. The incorporation of conjugate based Powell's method as local search algorithm in FPA increases the process of exploitation and thus it balanced in biased nature. PFPA shows higher convergence rate towards optimal solution and the computational results over other competitor algorithm for solving MEB. Embedment of Powell's method in FPA shows higher computational complexity and other existing exact methods to solve MEB in terms of time complexity.

6 Conclusion

This paper proposes a variant of Flower pollination algorithm based on Powell's method to solve minimum energy broadcast problem in wireless sensor network. The proposed algorithm is compared with other state-of-the-art heuristic existing algorithms.

- To solve MEB efficiently FPA is used as a global search and conjugated Powell's method is used as a local search. This incorporation balances the process of both explo-

ration and exploitation technique. The proposed PFPA was compared with three existing competitor algorithms using instances taken from DAG.

- The simulation results show the performance of the proposed algorithm reduces over existing algorithms. The empirical result validates the efficiency of the proposed algorithm to MEB problem in smaller and larger sized networks. Thus our algorithm does both deterministic and heuristic search towards obtaining optimal solution in solving MEB.

The future work of this research can be extended to

- adapting PFPA for solving various NP-hard combinatorial problems,
- further tuning the heuristic function and balancing exploration and exploitation process,
- investigating to apply in various coverage problems in wireless sensor networks,
- exploring unfeasible solution in the search space to emulate optimal solutions.

Acknowledgements This work is a part of the Research Projects sponsored by the Major Project Scheme, UGC, India, Reference Nos: F.No./2014-15/NFO-2014-15-OBC-PON-3843/(SA-III/WEBSITE), dated March 2015. The authors would like to express their thanks for the financial supports offered by the Sponsored Agency.

References

1. Al-Shihabi, S., Merz, P., & Wolf, S. (2007). Nested partitioning for the minimum energy broadcast problem. *International conference on learning and intelligent optimization* (pp. 1–11). Berlin Heidelberg: Springer.
2. Arivudainambi, D., & Rekha, D. (2013). Memetic algorithm for minimum energy broadcast problem in wireless ad hoc networks. *Swarm and Evolutionary Computation*, 12, 57–64.
3. Bauer, J., Haugland, D., & Yuan, D. (2009). A fast local search method for minimum energy broadcast in wireless ad hoc networks. *Operations Research Letters*, 37(2), 75–79.
4. Bekdaş, G., Nigdeli, S. M., & Yang, X.-S. (2015). Sizing optimization of truss structures using flower pollination algorithm. *Applied Soft Computing*, 37, 322–331.
5. Čagalj, M., Hubaux, J.-P., & Enz, C. (2002). Minimum-energy broadcast in all-wireless networks: NP-completeness and distribution issues. In *Proceedings of the 8th annual international conference on mobile computing and networking* (pp. 172–182). ACM.
6. Das, A. K., Marks, R. J., El-Sharkawi, M., Arabshahi, P., & Gray, A. (2002). The minimum power broadcast problem in wireless networks: An ant colony system approach. In *Proceedings of the IEEE workshop on wireless communications and networking*.
7. Das, A. K., Marks, R. J., El-Sharkawi, M., Arabshahi, P., & Gray, A. (2003). r-shrink: A heuristic for improving minimum power broadcast trees in wireless networks. In *Global telecommunications conference, 2003. GLOBECOM'03* (vol. 1, pp. 523–527). IEEE.
8. Guo, S., & Yang, O. W. W. (2007). Energy-aware multicasting in wireless ad hoc networks: A survey and discussion. *Computer Communications*, 30(9), 2129–2148.

9. Guo, S., & Yang, O. (2004). A dynamic multicast tree reconstruction algorithm for minimum-energy multicasting in wireless ad hoc networks. In *2004 IEEE international conference on performance, computing, and communications* (pp. 637–642). IEEE.
10. Hernández, H., & Blum, C. (2009). Ant colony optimization for multicasting in static wireless ad-hoc networks. *Swarm Intelligence*, 3(2), 125–148.
11. Hsiao, P.-C., Chiang, T.-C., & Fu, L.-C. (2013). Static and dynamic minimum energy broadcast problem in wireless ad-hoc networks: A PSO-based approach and analysis. *Applied Soft Computing*, 13(12), 4786–4801.
12. Jiang, J.-A., Chen, C.-P., Chuang, C. L., Lin, T.-S., Tseng, C.-L., Yang, E.-C., & Wang, Y.-C. (2009). CoCMA: Energy-efficient coverage control in cluster-based wireless sensor networks using a memetic algorithm. *Sensors*, 9(6), 4918–4940.
13. Kang, I., & Poovendran, R. (2003). A novel power-efficient broadcast routing algorithm exploiting broadcast efficiency. In *Vehicular technology conference, 2003. VTC 2003-fall. 2003 IEEE 58th* (vol. 5, pp. 2926–2930). IEEE.
14. Kang, I., & Poovendran, R. (2004). Broadcast with heterogeneous node capability [wireless ad hoc or sensor networks]. In *Global telecommunications conference, 2004. GLOBECOM'04* (vol. 6, pp. 4114–4119). IEEE.
15. Kang, I., & Poovendran, R. (2005). Iterated local optimization for minimum energy broadcast. In *Third international symposium on modeling and optimization in mobile, ad hoc, and wireless networks, 2005. WIOPT 2005* (pp. 332–341). IEEE.
16. Khalil, E. A., & Bara'a, A. A. (2011). Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks. *Swarm and Evolutionary Computation*, 1(4), 195–203.
17. Klasing, R., Navarra, A., Papadopoulos, A., & Pérennes, A. (2004). Adaptive broadcast consumption (ABC), a new heuristic and new bound for the minimum energy broadcast routing problem. *International conference on research in networking* (pp. 866–877). Berlin Heidelberg: Springer.
18. Konstantinidis, A., Yang, K., Chen, H.-H., & Zhang, Q. (2007). Energy-aware topology control for wireless sensor networks using memetic algorithms. *Computer Communications*, 30(14), 2753–2764.
19. Konstantinidis, A., Yang, K., Zhang, Q., & Zeinalipour-Yazti, D. (2010). A multi-objective evolutionary algorithm for the deployment and power assignment problem in wireless sensor networks. *Computer Networks*, 54(6), 960–976.
20. Liang, W. (2002). Constructing minimum-energy broadcast trees in wireless ad hoc networks. In *Proceedings of the 3rd ACM international symposium on mobile ad hoc networking and computing* (pp. 112–122). ACM.
21. Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. *Physical Review E*, 49(5), 4677.
22. Miller, C., & Poellabauer, C. (2009). A decentralized approach to minimum-energy broadcasting in static ad hoc networks. *International conference on ad-hoc networks and wireless* (pp. 298–311). Berlin Heidelberg: Springer.
23. Molina, G., & Alba, E. (2011). Location discovery in wireless sensor networks using metaheuristics. *Applied Soft Computing*, 11(1), 1223–1240.
24. Montemanni, R., Gambardella, L. M., & Das, A. K. (2005). The minimum power broadcast problem in wireless networks: A simulated annealing approach. In *Wireless communications and networking conference, 2005 IEEE* (vol. 4, pp. 2057–2062).
25. Pavlyukevich, I. (2007). Lévy flights, non-local search and simulated annealing. *Journal of Computational Physics*, 226(2), 1830–1844.
26. Powell, M. J. D. (1977). Restart procedures for the conjugate gradient method. *Mathematical Programming*, 12(1), 241–254.
27. Rajeswari, M., Amudhavel, J., Pothula, S., & Dhavachelvan, P. (2017). Directed bee colony optimization algorithm to solve the nurse rostering problem. *Computational Intelligence and Neuroscience*, 2017, 26. <https://doi.org/10.1155/2017/6563498>.
28. Raju, R., Amudhavel, J., Kannan, N., & Monisha, M. (2014). A bio inspired energy-aware multi objective chiropteran algorithm (EAMOCA) for hybrid cloud computing environment. In *2014 International conference on green computing communication and electrical engineering (ICGC-CEE)* (pp. 1–5). IEEE.
29. Reynolds, A. M., & Frye, M. A. (2007). Free-flight odor tracking in *Drosophila* is consistent with an optimal intermittent scale-free search. *PLoS ONE*, 2(4), e354.
30. Sengupta, S., Das, S., Nasir, M. D., & Panigrahi, B. K. (2013). Multi-objective node deployment in WSNs: In search of an optimal trade-off among coverage, lifetime, energy consumption, and connectivity. *Engineering Applications of Artificial Intelligence*, 26(1), 405–416.

31. Singh, A., & Bhukya, W. N. (2011). A hybrid genetic algorithm for the minimum energy broadcast problem in wireless ad hoc networks. *Applied Soft Computing*, 11(1), 667–674.
32. Wan, P.-J., Călinescu, G., Li, X.-Y., & Frieder, O. (2002). Minimum-energy broadcasting in static ad hoc wireless networks. *Wireless Networks*, 8(6), 607–617.
33. Wieselthier, J. E., Nguyen, G. D., & Ephremides, A. (2000). On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *INFOCOM 2000. Nineteenth annual joint conference of the IEEE computer and communications societies. Proceedings* (vol. 2, pp. 585–594). IEEE.
34. Wolf, S., & Merz, P. (2008). Evolutionary local search for the minimum energy broadcast problem. *European conference on evolutionary computation in combinatorial optimization* (pp. 61–72). Berlin Heidelberg: Springer.
35. Wu, X., Wang, X., & Liu, R. (2008). Solving minimum power broadcast problem in wireless ad-hoc networks using genetic algorithm. In *Communication networks and services research conference, 2008. CNSR 2008. 6th annual* (pp. 203–207). IEEE.
36. Yang, X.-S. (2012). Flower pollination algorithm for global optimization. *International conference on unconventional computing and natural computation* (pp. 240–249). Berlin Heidelberg: Springer.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Dr. M. Rajeswari received his M.Tech. and Ph.D. in Computer Science and Engineering from Pondicherry Central University, Puducherry, India. She has obtained her M.Tech. in Network and Internet Engineering and awarded University Gold Medal. Completed her B.Tech. in Computer science and Engineering from Pondicherry Central University. Currently, he is serving as an Assistant Professor at the Computer Science & Engineering Department in Sri Manakula Vinayagar Engineering College, Puducherry, India. Her research areas include Bio-Inspired Algorithms, Web Services, wireless sensor network, network security and Cloud Computing.



Dr. Kalaipriyan Thirugnanasambandam received his M.Tech. and Ph.D. in Computer Science and Engineering from Pondicherry Central University, Puducherry, India. Currently, he is serving as an Assistant Professor at the Computer Science & Engineering Department in KL University, Andhra Pradesh, India. His research interests are Bio Inspired Computing, Network Security, Ad-hoc Networks. The publication list includes National and International Journals, National and International Conferences.



Dr. R. S. Raghav received his M.Tech. and Ph.D. in Computer Science and Engineering from Pondicherry Central University, Puducherry, India. Currently, he is serving as an Assistant Professor at the Computer Science & Engineering Department in KL University, Andhra Pradesh, India. His research interests are Bio Inspired Computing, Network Security, Ad-hoc Networks, Cloud Computing and Big data analytics. The publication list includes National and International Journals, National and International Conferences.



U. Prabu is working as Assistant Professor in Department of Computer Science and Engineering, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, Andhra Pradesh. He is currently pursuing his Ph.D. at Department of CSE, Pondicherry Engineering College, Pondicherry. His research interest includes Software Defined Networks, Routing Algorithms, Optimization Algorithms and Information Retrieval. He is a Life Member in ISTE and CSI.




Devaraj Saravanan received the Bachelor of Technology degree from Anna University, Chennai, India, Master of Technology from Pondicherry University, Pondicherry, India, and Pursuing Ph.D. degree from the Pondicherry University, Pondicherry, India. His current research interests include Vehicular Networks, bio Inspired Algorithms, Internet of Things. He has co-authored more than 20 papers in Scopus and peer re-viewed journals.



Dr. Dinesh Kumar Anguraj is currently working as Associate Professor in the Department of Computer Science and Engineering at Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India. He received his Doctor of Philosophy in Information and Communication Engineering from Anna University, Tamil Nadu in 2018 and received his Master of Engineering in Computer Science and Engineering from Karpagam University, Tamil Nadu in 2013. His research interest includes Wireless Sensor Network, Body Area Network and Network Security. He is served as a reviewer for various International Journals. He has published many research articles in refereed journals and IEEE & Springer associated conferences.

Authors and Affiliations

M. Rajeswari¹ · Kalaipriyan Thirugnanasambandam² · R. S. Raghav³ · U. Prabu⁴ · Devaraj Saravanan⁵ · Dinesh Kumar Anguraj⁶ 

M. Rajeswari
raji.rajeswari18@gmail.com

Kalaipriyan Thirugnanasambandam
kalaip27@gmail.com

R. S. Raghav
vpmrags@gmail.com

U. Prabu
uprabu28@gmail.com

Devaraj Saravanan
svsaravana1@gmail.com

¹ Sri Manakula Vinayagar Engineering College, Madagadipet, Puducherry, India

² Department of Computer Science and Technology, School of Computers, Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh, India

³ School of Computing, SASTRA Deemed University, Thanjavur, Tamilnadu, India

⁴ Department of CSE, Velagapudi Ramakrishna Siddhartha Engineering College, Vijayawada, India

⁵ Department of Computer Science Engineering, VIT Bhopal University, Kothri Kalan, Sehore, Madhya Pradesh, India

⁶ Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Greenfields, Vaddeswaram, Guntur, Andhra Pradesh, India