Advances and Applications in Mathematical Sciences
Volume 17, Issue 1, November 2017, Pages 51-64

MOTH-FLAME ALGORITHM FOR SOLVING COVERAGE PROBLEM IN WIRELESS SENSOR NETWORK

M. RAJESWARI[1], J. AMUDHAVEL[2] and P. DHAVACHELVAN[3]

[1,3]Department of Computer Science
Pondicherry University
Puducherry, India
E-mail: raji.rajeswari18@gmail.com
dhavachelvan@gmail.com

[2]Department of Computer Science and Engineering
KL University
Andhra Pradesh, India
E-mail: info.amudhavel@gmail.com

Abstract

Optimization is the process of finding better results with respect to time which is one of the most desirable features in recent technology. WSN is one among the domain which utilizes the optimization process in the field of sensor node deployment, node localization, message routing, etc. Moth-Flame optimization algorithm is a bio-inspired optimization method which is used in this paper to solve *k*-coverage node deployment on target based WSN. The proposed methodology in nature moths navigation methods is transverse orientation and moths maintain a fixed angle. The main objective of this Moth-Flame algorithm is going to solve the convergence problem to minimization of nodes and maximize coverage in a wireless sensor network. The experimental results are very effective and achieve the high performance.

1. Introduction

In WSN technology the Optimization process is used for finding the finest possible solution (position of nodes) for a given network problem. In WSN world, many problems like deployment of node, shortest path and energy related issues can be viewed as optimization problems. So we need effective and simple new optimization techniques for the solving difficulties of

2010 Mathematics Subject Classification: 68N13, 97N50.
Keywords: Moth-Flame algorithm, evolutionary algorithm, coverage problem, wireless sensor network, optimization problem.
Received March 10, 2017; Accepted July 20, 2017

problems in the dynamic changing environment [8]. Over past several decades, some kinds of techniques have been planned to solve optimization problems in WSN environment. For instance, WSNs technology is implemented in many fields for monitoring, tracking and security purpose. Here the WSN should consider an important factor which plays the vital role in every WSN, known as coverage. It is one of the key performance metrics for Wireless Sensor Networks (WSNs), where it illustrates the movement of the sensor and it also explains how the sensor shows its full potential for monitoring. The Area coverage or target coverage problems are two main issues in WSN. The target coverage algorithms are used to increase the number of targets nodes, which can coverage to all target nodes in the sensing field. The area coverage algorithms are used to enhance the covered area of the complete sensing field. The deployment of the sensor node in the wireless environment is difficult to access because the wireless doesn't have any fixed topology or position of nodes to have better network coverage. Here, we use a new meta-heuristic optimization method known as Moth-flame optimization (MFO). It imitates the movement flow of moths in nature known as transverse orientation. In this algorithm, moths and flames define the solutions. It is proved that this algorithm has the ability to show very better results compared with other meta-heuristic optimization algorithms [8-12, 19, 22]. This algorithm can be used for solving the coverage related issues in sensor environment. Energy aware technology used in Cloud environment [20, 26].

In this work, the proposed algorithm helps solve coverage problem in WSN by deploying sensor nodes within its transmission region. Here, we use Moth-Flame Optimization algorithm for handling coverage related issues. The remainder sector of the work is organized as follows. Section 2 summarizes the related work in evolutionary algorithm. Sections 3 explain in detail about the problem formulation of the coverage problem. Section 4 describes the flow of Moth-Flame optimization algorithm. Section 5 shows the empirical result on solving coverage problem using Moth-Flame algorithm. Finally, section 6 concludes the work.

2. Literature Study

Mirjalili Seyedali, discuss about a novel metaheuristic optimization algorithm, known as Moth-Flame Optimization (MFO) algorithm [1]. The main objective of this optimizer is to carry the exploration technique of moths defined as called transverse orientation. Moths fly in dark by managing a fixed angle with respect to the moon. The moth can travel in a straight line for long distances which shows the movement of the very effective mechanism for traveling. They also have the tendency to move spiral path around artificial lights. This paper defines the mathematical process of the Moth-Flame behavior to achieve optimization. In the result is evaluated and it is compared with other inspired algorithms. In the domain of optimization some of the other algorithms include Tabu Search [14], Hill Climbing [15], Iterated Local Search [16], Differential Evolution [9], Evolutionary Strategy [10], Optimization based on Artificial Intelligence [11], Evolutionary Programming [12], etc. A number of bio inspired algorithms are proposed for solving optimization problems are based on no free lunch theorem [13]. Bioinspired algorithm is used to solve scheduling problem [27].

Zhang, Li, et al. [2] propose a facial expression recognition system, by using the variant of evolutionary moth fire-fly algorithm. It has a modified Local Binary Pattern descriptor for providing an initial discriminative face representation. The proposed evolutionary fire-fly algorithm explains the exploration process carried by the moths in the spiral search behavior. It also describes the searching process of fire flies to have the convergence of the Levy flight fire-fly algorithm (LFA) and the Moth-Flame optimization (MFO) algorithm [4]. It indicates the spiral search movement of the moths to improve local exploitation of the fireflies. It also has the ability to represent the best solutions identified by the moths and it used as search agents to increase global exploration.

Li, Zhiming, et al. [3] discuss the moth-flame optimization (MFO) algorithm to show its slow convergence and low precision rate. In this work, they proposed an enhanced version of MFO algorithm based on Levy-flight strategy, which defined as LMFO. The nature of Levy-flight is diversity of the population is increased against premature convergence, in order to make the algorithm moving out from local optimum more effectively. This proposed approach is used for getting a better trade-off between exploration and exploitation ability of MFO. It shows the proposed LMFO can be quick and more robust when compared with MFO.

Silvestri, Simone and Ken Goss [5] propose a Mobi Bar a mobile sensor for self-governing deployment algorithm for k-barrier coverage. The proposed algorithm organizes the migration of sensor movements to develop k distinct complete barriers and it helps to provide information about redundancy level. The maximum coverage level of barrier with required sensors can be achieved by removing in a finite time. Furthermore, the Mobi Bar act as a automatic-reconfigure and self-heal the network, in order to solve the sudden failures with dynamic coverage.

Ab Aziz, Nor Azlina [6] discuss an algorithm to improvise coverage of the network by moving the sensor nodes, to reach the target sensor node, the network need an optimal position to place their nodes. However the migration of sensors nodes consumes high energy level. Optimization algorithm is used in web services to optimize based on QoS metrics [23-25]. In this paper three algorithms are shown to optimize the coverage of WSN. The proposed algorithms are based on particle swarm optimization (PSO) [7], which has the ability to give better performance record. It shows how the proposed algorithms place the nodes in a better place to solve the coverage problem in dynamic environment.

3. Problem Formulation

The coverage problem a set of pre fixed potential positions and a set of target points; here they have to place sensor nodes to select minimum number of potential positions to full fill the targets in the sensor networks. In wireless sensor networks the communication range of a sensor node covered a target and sensor node able to communicate with the base station directly in the sensor nodes.

Let us illustrate all the terminologies used in the problem are described. A set of target points TP represent as $TP = \{t_1, t_2, t_3, \dots t_n\}$. A set of given potential positions PP can be represent as $PP = \{p_1, p_2, p_3, \dots, p_k\}$. The N_{com} represents the communications range of the sensor nodes. The sensing range of the sensor nodes can be denoted as N_{sen}. $\text{dist}(t_i, s_j)$ denotes the distance between t_i and s_j. A set of sensor nodes $Cov(t_i)$, which are within sensing range of t_i. In other words, t_i is covered by $Cov(t_i)$.

$$Cov(t_i) = \{s_j \mid dist(t_i, s_j) \le N_{sen}, \forall j, 1 \le j \le M\}. \tag{1}$$

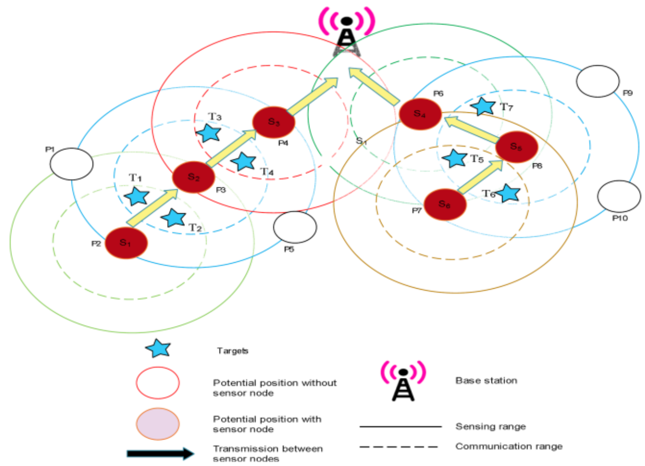


Figure 1. Coverage connected network.

An example, in a wireless sensor networks they initiated a target based networks with position and targets. They take as 10 samples for potential positions $\{p_1, p_2, p_3, \ldots, p_{10}\}$ and seven targets $\{t_1, t_2, t_3, \ldots t_7\}$ is shown in figure. Out of 10 potential positions the sensor nodes are placed on selected six potential positions. In the Figure 1 the target t_2 is covered because this sensor node within the communication range placed at potential position p_2 and this node is communicate with the base station via the placed sensor nodes at the potential positions are p_3 and p_4.

4. Moth-Flame Optimization Algorithm

Seyedali Mirjalili [1] proposed a Moth-Flame Optimization algorithm which inspired from the navigation methods of moths called as transverse orientation. In this algorithm Moth is the individuals and a set of moth is called as population. And flame is said to be the best solution for each moth. It comes under the concept like PSO. In PSO there exist ‘p-best’ which is said to be the personal best of each particle. Likewise, this moth-flame contains one flame per moth which is considered to be as its best position. During the iteration this flame will get updated if there exist any better solution.

Algorithm 1: Moth-Flame

```
Initialize the parameters for Moth-flame
Initialize Moth position M_i randomly
For eachi = 1: n do
Calculate the fitness function f_i
End for
While(iteration ≤ max _iteration)do
   Update the position of M_i
   Calculate the number of flames using Eq.(5)
   Evaluate the fitness function f_i
   If(iteration == 1)then
      F = sort(M)
            OF = sort(OM)
Else
            F = sort(M_{t-1}, M_t)
           OF = sort(M_{t-1}, M_t)
End if
For eachi = 1: n do
For eachj = 1: d do
         Update the values of r and t
         Calculate the value of D respect to its
corresponding moth using Eq.(4)
         Update M(i, j) respect to its
corresponding moth using Eq.(3)
End for
   End for
End while
Print the best solution
```

Figure 2. The pseudo code of Moth-Flame Optimization algorithm.

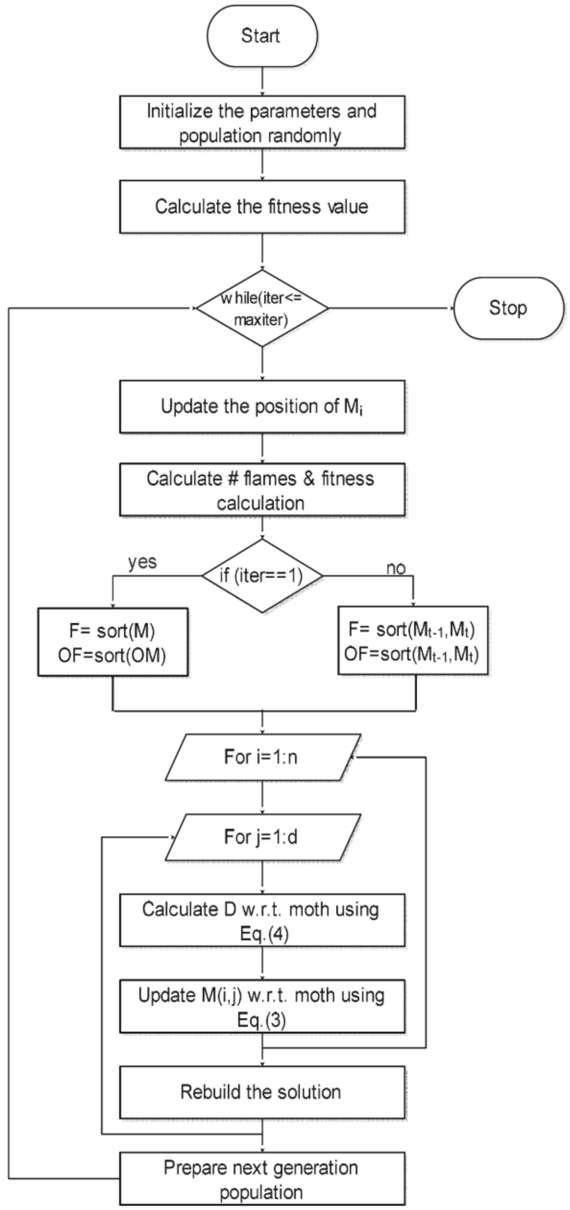


Figure 3. Flowchart of Moth-Flame Optimization Algorithm.

Moth-Flame algorithm consists of Moth (*M*), their respective fitness values (*OM*), Flames (*F*) which is considered as their previous best position, of their respective fitness value.

The equation to find the best optimal value is given as

$$M_i = S(M_i, F_j), \tag{2}$$

where M_i indicates i^{th} moth, F_j indicates j^{th} Flame and S indicates the spiral function of the moth.

The spiral function of the moth is given as

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j, \tag{3}$$

where D_i indicates the distance between the j^{th} Flame and i^{th} moth.

$$D_i = | F_j - M_i |. \tag{4}$$

The moth can be able to converge or exploit in the given search space by changing the value t. when the position of a moth gets changed then it indicates that there comes the exploitation. Unfortunately, the author did not add any heuristic function or meta-heuristic function for exploration. An adaptive mechanism to solve number of flames during each iteration is mathematically defined as,

$$\vartheta = \text{round}\left(N - 1 * \frac{N-1}{T}\right), \tag{5}$$

where N is the maximum number of flames, l is the number of current iteration, and T is the maximum number of iterations.

The pseudo code for the Moth-Flame algorithm is shown in Figure 2 and the work flow of Moth-Flame algorithm is represented in Figure 3.

5. Experimental Results

The proposed methodology for solving m-connected k-coverage optimization problem in Wireless Sensor Networks has been implemented in MATLAB 8.3 in the system configuration of Intel Cor i7 Processor with 3.2GHz speed and 4GB RAM. For testing the efficiency of proposed algorithm, two different test bed designs are prepared. One with a grid of 50×50 meters and another with 100×100 grid square meter. Positions of the sensors and generated randomly by fixing the targets in prior. Three performance metrics are used to evaluate the performance of the proposed algorithm and the proposed algorithm is compared with two existing approaches namely Genetic Algorithm [18] and Particle Swarm Optimization [17]. The performance metrics are computational time, F value and total number of sensors deployed for efficient coverage of targets.

The Parameter settings for environment configuration is shown in Table 1 as follows:

Table 1. Configuration Parameter.

Type	**Method**
Population size	100
Maximum iterations	500
R	3
C	10
Threshold Value (∂)	0.5

5.1. Performance Metrics

5.1.1. Computational Time

Computational time is defined as the total time taken to complete the runtime of proposed algorithm. Best solution is not set as an epoch value for termination of algorithm since this evaluation is purely based on random location.

$$\text{Computational Time} = \text{Runtime of proposed algorithm} \tag{6}$$

5.1.2. No. of Sensor Nodes Deployed

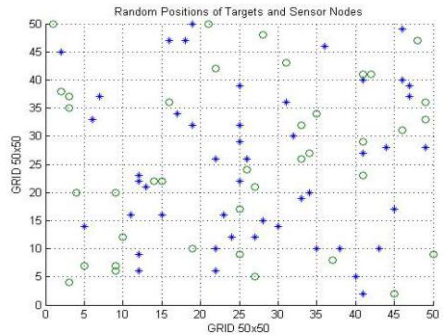
Number of nodes that are placed in the grid to cover all the targets located in the simulated area.

5.1.3. *F* value

F value is the ratio between the number of available positions to plot the sensor nodes and the total number of sensor node deployed at the end of the cycle.

$$F = \frac{\# TP}{\# AP}, \tag{7}$$

where *AP* is the allocated positions with sensor nodes and *TP* total number of available positions.



o-Sensor Nodes ∗-Targets

Figure 4. MALTLAB Simulation region for 50×50.

The required number of targets and sensor nodes that are randomly generated in 50×50 grid in MATLAB 8.3 is shown in Figure 4. The results for 50×50 grid is tabulated in Table 2. With the performance results of proposed algorithm on coverage problem, other two existing approaches namely *GA* and PSO are tabulated for comparing the efficiency of proposed algorithm.

Table 2. Experiment Result of Ga Vs Pso Vs Moth-Flame For 50×50.

	Comp. Time (*s*)	# Nodes Deployed	*F*-Value
GA	7.01	36	1.11
PSO	6.61	28	1.42
Moth-Flame	5.79	22	1.81

From Table 2, the results of *GA*, PSO and the proposed Moth-Flame Algorithm are tabulated for the given performance metrics. On comparing the results in terms of computational time, out proposed algorithm simulation time is higher than the PSO but when compared with *GA* it is less. In terms of number of nodes deployed and *F*-value our proposed algorithm shows significant improvement over existing approaches.

The convergence of total number of nodes deployed in the given grid 50×50 on every 100 iterations are given in Figure 5.

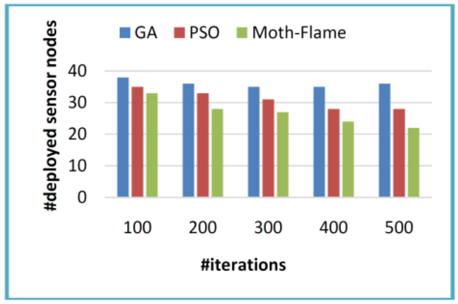
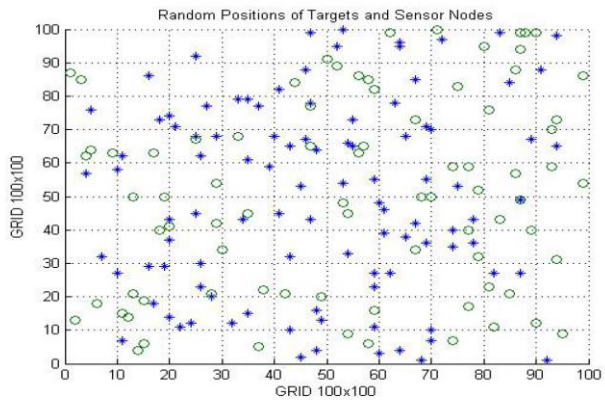


Figure 5. Comparison of convergence for 50×50.



o-Sensor Nodes $*$-Targets

Figure 6. MALTLAB Simulation region for 100×100.

The required number of targets and sensor nodes that are randomly generated in 100×100 grid in MATLAB 8.3 is shown in Figure 6 as follows. The results for 100×100 grid are tabulated in Table 3. With the performance results of proposed algorithm on coverage problem, other two existing approaches namely GA and PSO are tabulated for comparing the efficiency of proposed algorithm. The simulation results for 100×100 grid are tabulated in Table 3.

From Table 3, the results of GA, PSO and the proposed Moth-Flame Search Algorithm are tabulated for the given performance metrics. In terms of number of nodes deployed and F-value our proposed algorithm shows significant improvement over existing approaches. For fair comparison, a detailed picture on performance of proposed algorithm is given in Figure 7.

Table 3. Experiment Result of Ga Vs Pso Vs Moth-Flame For 100×100.

	Comp. Time (s)	# Nodes Deployed	*F*-Value
GA	11.25	60	1.33
PSO	9.38	54	1.48
Moth-Flame	8.79	50	1.60

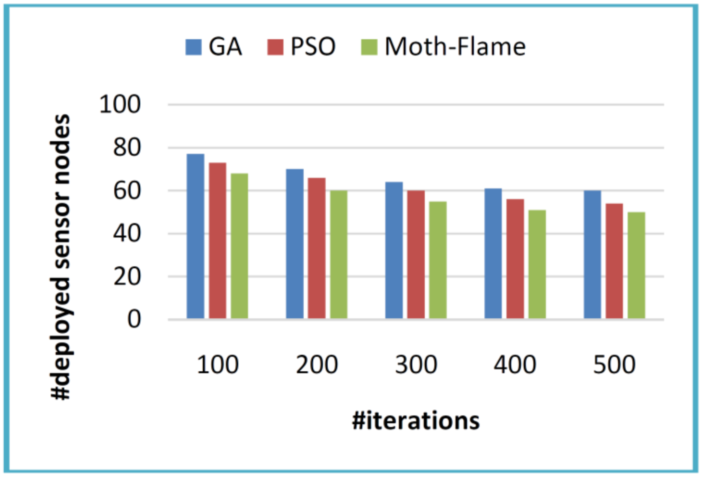


Figure 7. Comparison of convergence for 100×100.

5. Conclusion

Moth-flame optimization algorithm is implemented for minimizing the number of sensor nodes to be deployed in WSN scenario. The proposed method shows a significant improvement in the experimental results when compared with the existing algorithms. The proposed method is high in terms of *F*-Value and number of sensor nodes deployed. The proposed methodology registers high level computational time when compared with PSO. Further investigation on parameter values and other testing feature on different dimension dataset and various research areas can be used in future research.

Acknowledgment

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] Mirjalili and Seyedali, Moth-flame optimization algorithm: A novel nature-inspire dheuristicparadigm, Knowledge-Based Systems 89 (2015), 228-249.

[2] Zhang Li, et al., Intelligent facial emotion recognition using moth-fire-fly optimization, Knowledge-Based Systems 111 (2016), 248-267.

[3] Li, Zhiming, et al, Lévy-Flight Moth-Flame Algorithm for Function Optimization and Engineering Design Problems, Mathematical Problems in Engineering 2016 (2016).

[4] Soliman, M. A. Ghada, Motaz M. H. Khorshid and Tarek H. M. Abou-El-Enien, Modified mothflame optimization algorithms for terrorism prediction,

[5] Silvestri, Simone and Ken Goss, Mobi Bar: An autonomous deployment algorithm for barrier coverage $w\ i^{th}$ mobile sensors, Ad Hoc Networks 54 (2017), 111-129.

[6] Ab Aziz and Nor Azlina, Wireless sensor networks coverage-energy algorithms based on particle swarm optimization, Emirates Journal for Engineering Research 18(2) (2013), 41-52.

[7] R. C. Eberhart and J. Kennedy, A new optimize rusing particles warm theory Proceedings of the Sixth International Symposiumon Micro Machine and Human Science (1995), 39-43.

[8] M. Colorni, V. Dorigo and Maniezzo, Distributed optimization by antcolonies, Proceedings of the First European Conference on Artificial Life (1991), 134-142.

[9] R. Storn and K. Price, Differential evolution as simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Opt. 11 (1997), 341-359.

[10] Rechenberg, Evolution strategy: optimization of technical systems by means of biological evolution, Fromman-Holzboog, Stuttgart 104 (1973).

[11] L. J. Fogel, A. J. Owens and M. J. Walsh, Artificial intelligence through hsimulated evolution, 1966.

[12] X. Yao, Y. Liu and G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999), 82-102.

[13] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput. 1 (1997), 67-82.

[14] F. Glover, Tabu search-Part I, ORSA J. Comput. 1 (1989), 190-206.

[15] L. Davis, Bit-Climbing, Representational bias, and test suite design, ICGA (1991), 18-23.

[16] H. R. Lourenço, O. C. Martin and T. Stutz, Iterated local search, 2001. Available from arXiv: preprint math/0102188.

[17] J. Kennedy and R. Eberhart, Particle swarm optimization, 1995 Proceedings of IEEE International Conference on Neural Networks; 1995.

[18] J. H. Holland, Adaptation in natural and artificial systems, MIT Press; (1992), 211.

[19] R. Raju, J. Amudhavel, Nevedha Kannan and M. Monisha, A bioinspired Energy-Aware Multi objective Chiropteran Algorithm (EAMOCA) for hybrid cloud computing genvironment, In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on, 1-5. IEEE, 2014.

[20] R. Raju, J. Amudhavel, N. Kannan and M. Monisha, Interpretation and evaluation of various hybrid energy aware technologies in cloud computing environment-*A* detailedsurvey, Proceedings of International Conferenceon Green Computing Communication and Electrical Engineering, ISBN: 978-1-4799-4982-3, pp.1-3, March 6th-8th, Coimbatore, India, 2014.

[21] R. Raju, J. Amudhavel, M. Pavithra, S. Anuja and B. Abinaya, A heuristic fault tolerant Map Reduce framework for minimizing make span in Hybrid Cloud Environment, In Green Computing Communication and Electrical Engineering (ICGCCEE), 2014 International Conference on, pp. 1-4. IEEE, 2014.

[22] J. Amudhavel, N. Mohanarangan, P. Dhavachelvan, R. Baskaran, M. S. Saleem Basha, A. K. Rehana Begum and K. Prem Kumar, A krill herd optimization based fault tolerance strategy in MANETs for dynamic mobility, In Circuit, Power and Computing Technologies (ICCPCT), 2015 International Conference on, pp. 1-7. IEEE, 2015.

[23] J. Amudhavel, U. Prabu, P. Dhavachelvan, N. Moganarangan, V. Ravishankar and R. Baskaran, Non-homogeneous hidden Markov model approach for load balancing in web server farms (NH 2 M 2-WSF), In Communicati on Technologies (GCCT), 2015 Global Conference on, pp. 843-845. IEEE, 2015.

[24] M. Rajeswari, G. Sambasivam, N. Balaji, M. S. Saleem Basha, T. Vengattaraman and P. Dhavachelvan, Appraisal and analysis on various web service composition approaches based on QoS factors, Journal of King Saud University-Computer and Information Sciences 26(1) (2014), 143-152.

[25] J. Amudhavel, U. Prabu, N. Saravanan, P. Dhavachelvan, R. Baskaran, and V. S. K. Venkatachalapathy, A Distributed Spanning Tree-Based Dynamic Self-Organizational Framework for Web Server, In Proceedings of the Second International Conference on Computer and Communication Technologies, pp. 333-339. Springer India, 2016.

[26] J. Amudhavel, U. Prabu, P. Dhavachelvan, N. Moganarangan, V. Ravishankar and R. Baskaran, A comprehensive analysis and performance assess menton QoS aware web services election algorithms, In Communication Technologies (GCCT), 2015 Global Conference on, pp. 846-849. IEEE, 2015.

[27] M. Rajeswari, J. Amudhavel, Sujatha Pothula and P. Dhavachelvan, Directed Bee Colony Optimization Algorithm to Solve the Nurse Rostering Problem, Computational Intelligence and Neuroscience, vol. 2017, Article ID 6563498, 26 pages, 2017. doi:10.1155/2017/6563498.