

“

---

---

## Optimization of Wireless Sensor Nodes Placement Using Particle Swarm Optimization Algorithm for Fire Detection

By

**A. Ameh \*, A. B. Salihu, M. David, H. Bello-Salau, and P. Enenche**

Department of Telecommunication Engineering,  
School of Electrical Engineering Technology,  
Federal University of Technology Minna, Nigeria

Email: \*[austyc123@gmail.com](mailto:austyc123@gmail.com)\*

---

---

### ABSTRACT

*The development in wireless communications and electronics brought about unprecedented advancement in Wireless Sensor Network (WSN). WSN consists of a group of sensor nodes that can communicate with each other through wireless medium and have at least a sink node where the data will be sent to. One important problem in implementing a WSN is the position of the nodes that will meet the design specifications such as coverage and connectivity. Hence, it is important that in sensor nodes deployment strategies, consideration should be given to the sensing range of sensors and how well the sensors can communicate with each other. Coverage and Connectivity are two important performance metrics in wireless sensor networks that show how a sensor field is monitored by the deployed wireless sensor nodes. Optimizing the position of sensor nodes placement can help in the realization of objective function of generating high throughput, low latency and fast convergence rate. They are quite a number of optimization algorithms that can be used to achieve this but the choice of an optimization algorithm depends on its suitability for the task when compared to others. Particle swarm optimization algorithm was used for this work because it is easy to implement, makes use of few parameters, fast convergence and better solution. Also, dijkstra shortest path algorithm was used in routing data through the shortest path to the sink.*

**Key words:** Wireless Sensor Networks, Particle Swarm Optimization, Dijkstra Algorithm, Throughput, Coverage, Connectivity, Deployment, Sink, Node

### INTRODUCTION

Wireless Sensor Networks (WSNs) consist of a group of sensor nodes that monitor the environment. The sensors obtain the data collected from the monitored environment and send them to a destination called sink within the sensor network. WSNs give a powerful combination of dispersed sensing, computing and communication. The ever increasing capabilities of these tiny sensor nodes, which comprises of sensing, data processing, and communicating, paves way for the realization of WSNs based on the collaborative efforts of other sensor nodes [1].

WSNs are made up of well distributed autonomous sensor nodes working together for numerous sensing and monitoring services. The introduction of low power embedded systems and wireless networking, brought about enhanced possibilities for distributed sensing applications. These technologies created an avenue for the implementation of wireless sensor networks, allowing easily configured, flexible sensors to be placed almost anywhere, and the data collected by the sensors transported over large distances through wireless networks [2].

Wireless sensor nodes make use of sensors to acquire data from the surroundings. These are hardware devices that respond to physical changes like temperature, smoke, pressure etc. Sensors take physical readings of the parameter to be monitored and have specific characteristics such as accuracy, sensitivity etc. WSN consists of small embedded sensor which interfaces and communicates with sensors through short range wireless transmitters [3].

## REVIEW OF RELATED WORKS

Several proposals have been made to help in early detection of an event. [4] Proposes the monitoring variation in temperature using a single sensor with an immediate audio-visual emergency response interface aimed at enhancing fire detection and the evacuation procedure.

[5] Propose an efficient, Internet of Things for an early fire event detection in a residential building. [6] Emphasized on the importance of using wireless sensor network for early forest fire detection. [7] Emphasized on early fire detection in the Mediterranean regions.

[8], Vicente offered an efficient system to detect smoke early. [9] Proposed and developed an algorithm for wildfire detection. [10] Used wireless sensor network to propose a fuzzy logic approach for early fire detection.

[11] Made an improvement in the system proposed by [7] with more accuracy. MATLAB was used for simulation. Wireless sensors for some times now been used for event detection [12]. The work

proposed in [13] made use of wireless sensor network for early forest fire detection. [14] proposed the classification of mobile sink as; Predictable, Random and Controlled. [15] Proposes the use of mobile sink in efficient data collection.

Quite a lot of works have been done on wireless sensor node placement. In [16], Lin proposed improvement on quality of service for a higher throughput. [17] Propose use of genetic algorithm for sensor nodes placement problem. [18] Proposed an improvement over the conventional Particle Swarm Optimization (IPSO) algorithm to optimize sensor nodes for better result.

## METHODOLOGY

The improvement in the field of wireless communications and electronics brought about huge expansion in Wireless Sensor Network (WSN). WSN is made up of sensor nodes and these nodes communicate using wireless medium and contain sink(s) node where the collected data are received and processed. WSN performs two important functions which are sensing and communication among various sensors [19].

### *Sensor Placement*

One major problem in the implementation of WSN is the position of the sensor node to achieve the aim of efficient coverage and connectivity. Effective coverage and connectivity of the sensor nodes eventually leads higher data throughput in WSN. Figure 1 shows sensor nodes placement scheme, the method, objective and roles.

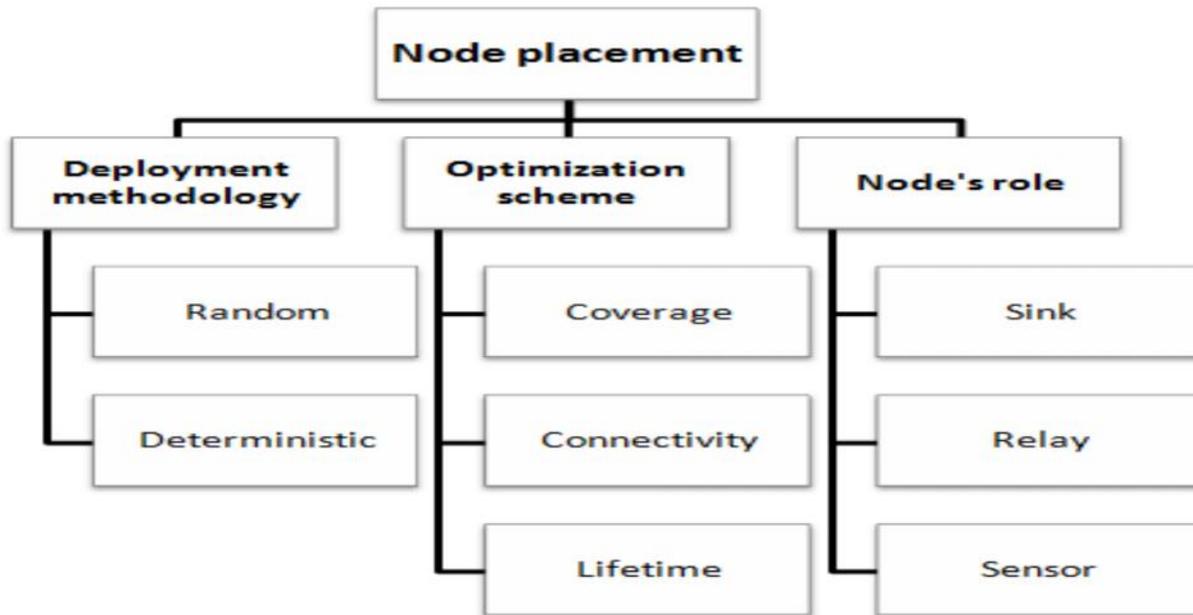


Figure 1 Sensor node placement methodologies

Coverage determines the sensing range of the sensor nodes and ensures that the region of interest is properly covered by the deployed sensors. The accuracy of the system relies on the quality of coverage within the monitored region [20]. The sensor nodes in the monitored region should maintain good communications (connectivity) with one another and with the sink(s). Hence, coverage and connectivity are the design parameters that should be considered concurrently in the deployment of a WSN.

### Particle Swarm Optimization (PSO)

Particle Swarm Algorithm is used to simulate a non-linear optimization problem. It is a population based stochastic optimization scheme inspired by the behavioural pattern of birds and fishes. A bird usually referred to as particle in this case represents a possible solution to an optimization problem and all particles have fitness values which show how close a particle is to the solution and also how close a swarm is to the solution. It is an algorithm that is simple and easy to implement.

Many optimization algorithms exist in literatures, however, the use of PSO

algorithm was considered particularly for its fast convergence and efficient performance, use of few parameters, ease of parallelizing for concurrent processing and simplicity [21][22][23]. It is an optimization technique that optimizes a problem by iteratively trying to improve a candidate solution, with regards to a given measure of quality.

### Dijkstra Algorithm

This is for single source shortest path problem. In a network, a shortest path should be used in routing data to the sink to minimize latency. This can be done via direct path or multi hops. The data has to find shortest path to the sink hence a minimization problem and a minimization issue is an optimization problem. Optimization problem can be solved using greedy method; greedy method says that problem should be solved in stages by taking one step at a time and considering one input at a time to get optimal solution. Greedy methods use predefined procedures and the procedures are to be followed to get optimal solution. Therefore Dijkstra Algorithm gives us procedure to get optimal solution by

following the shortest path in routing data from source nodes to the sink.

Describing how Dijkstra Algorithm work using simple diagram below; from the diagram there is a direct path from 1 to 2 but no direct path from 1 to 3 therefore

represented with infinity. Distance from 1 to 2 is 2 and distance from 1 to 3 via multi hop is 6 which is less than infinity hence the distance from 1 to 3 is 6 which is the shortest path. This is shown in Figure. 2.

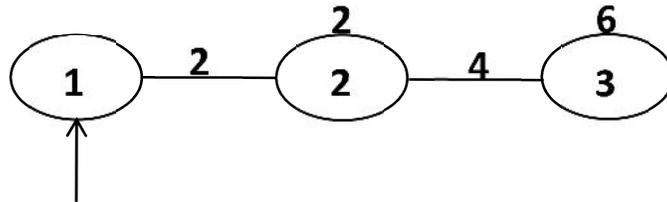


Figure 2 Dijkstra shortest path techniques

**System Model**

The steps followed in modeling the network is shown detailed in the following steps.

The modelling of the network in MATLAB followed the following steps:

- (1) Create a Function for a grid wireless sensor network of 500x300 meters field and add sensors to it with initial random positions each within the grid.
- (2) Create a Function to simulate the network using multi-hop/unicast with Dijkstra algorithm for routing, based on shortest path to the sink.

(3) Create a Receive/transmit matrix (RxTxM) for all nodes and update it as messages are sent and received by the nodes.

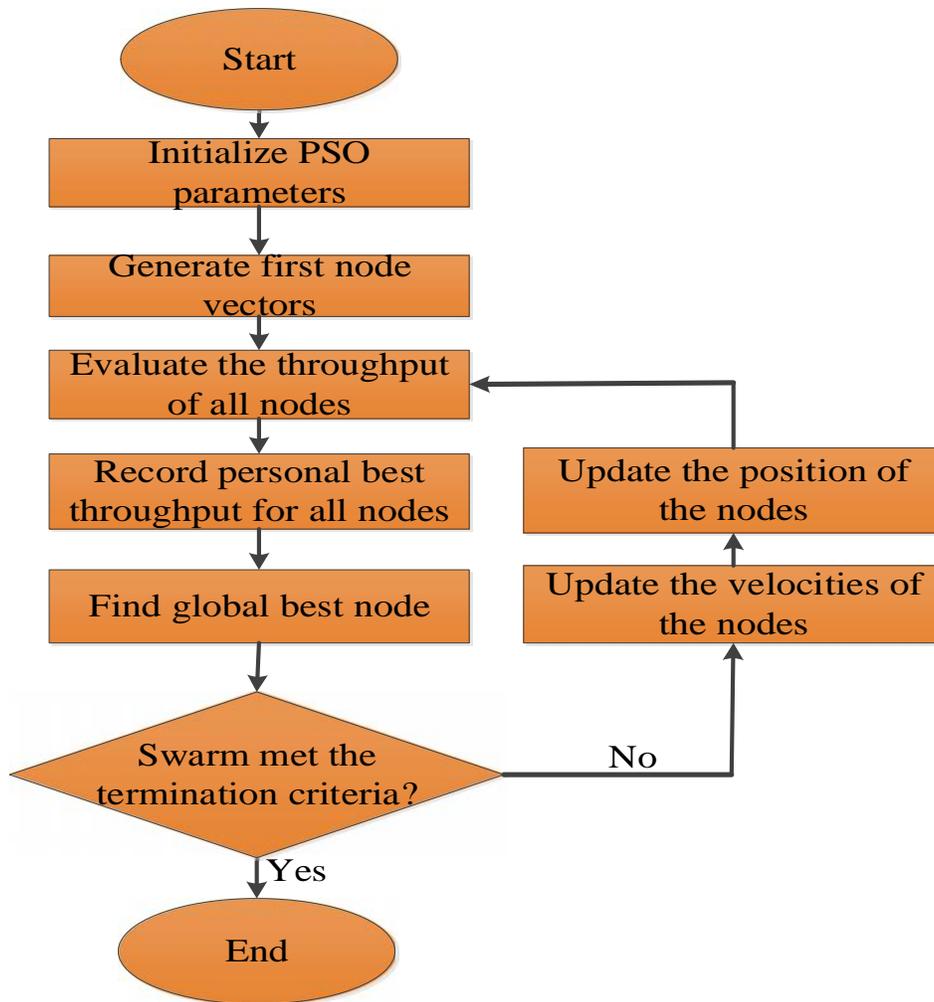
- (4) Compute the Throughput of the Network
- (5) Integrate Particle Swarm Optimization (PSO) Algorithm, to find the best nodes and sink positions that return the maximum throughput.

**Simulation Parameters**

The parameters used for the simulation is shown in Table 1, the flow diagrams for PSO and Dijkstra Algorithms are shown in Figure 3 and 4 respectively.

Table 1: Simulation Parameters

S/No	Parameters	Value
1	Number of Node	3, 6, 9, 12, 16
2	Number of population (n <sub>pop</sub> )	2, 4, 6, 8, 10
3	C <sub>1</sub>	1
4	C <sub>2</sub>	2
5	Inertia weight	1
6	Optimization Algorithm	PSO
7	Routing Algorithm	Dijkstra
8	Number of Sinks	1



**Figure 3** The Particle Swarm Algorithm Flow chart

This can be literally explained thus:

1. Initialize the size of the PSO parameters
2. Generate vector for the first node
3. Evaluate throughput/fitness value of all the nodes
4. Record personal best throughput/fitness values for all the nodes
5. Find the global best nodes as the overall best
6. Update the velocity of the nodes
7. Update the position of the nodes

The parameters are explained below:

- $x_k^i$  - The position of the node
  - $v_k^i$  - Velocity of the node
  - $p_k^i$  - Individual node best position
  - $p_k^g$  - Global nodes best position
  - $c_1, c_2$  - Nodes learning factors
  - $r_1, r_2$  - Random numbers between 0 and 1
- The position of individual node is updated as follows:
- $$x_{k+1}^i = x_k^i + v_{k+1}^i,$$
- The velocity of the node is calculated as follows:
- $$v_{k+1}^i = v_k^i + c_1 r_2 (p_k^i - x_k^i) + c_2 r_2 (p_k^g - x_k^i).$$

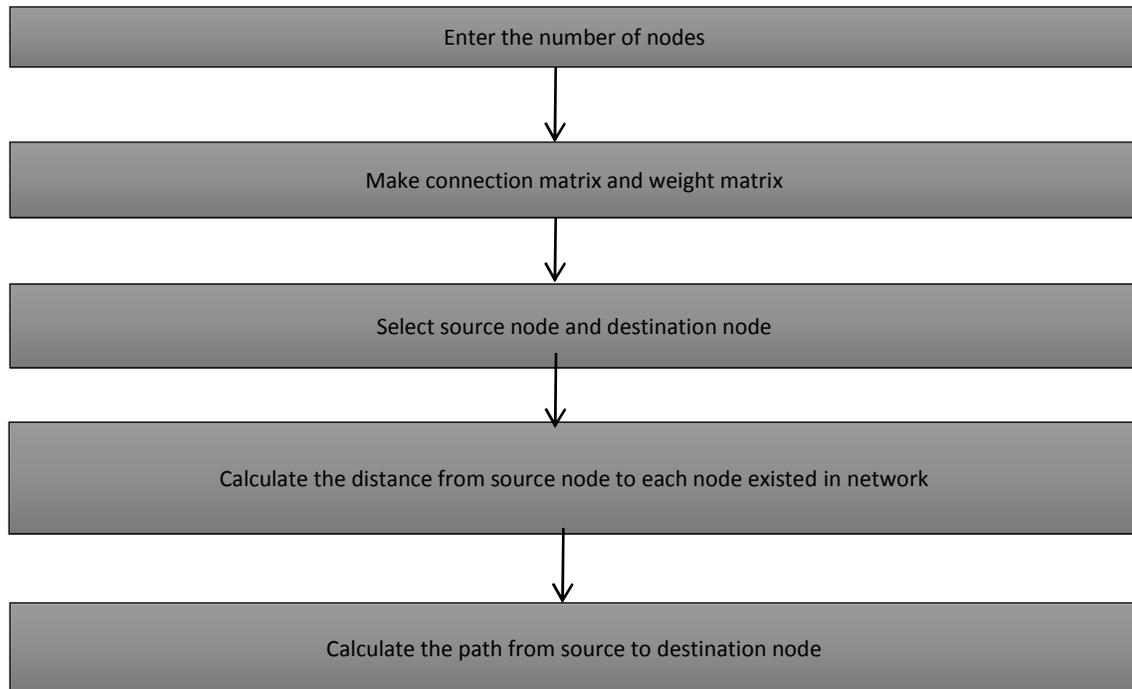


Figure 4 Dijkstra Flow Chart of Algorithm

**RESULT AND DISCUSSION**

**Model of sensor node placement as an optimization problem**

In order to model the node placement problem into an optimization problem using PSO, the following need to be done.

**Particle representation**

The nodes are represented as particles in the search space. Each node is represented by the x-y coordinates of their location in the search space. Each particle is represented as shown in Table 2, for a 9-node problem.

Table 2 Dimensions of a 9-node problem

1	2	3	4	5	6	7	8	9	.....	.....
N <sub>x1</sub>	N <sub>x2</sub>	N <sub>x3</sub>	N <sub>x4</sub>	N <sub>x5</sub>	N <sub>x6</sub>	N <sub>x7</sub>	N <sub>x8</sub>	N <sub>x9</sub>	.....	N <sub>xm</sub>
N <sub>y1</sub>	N <sub>y2</sub>	N <sub>y3</sub>	N <sub>y4</sub>	N <sub>y5</sub>	N <sub>y6</sub>	N <sub>y7</sub>	N <sub>y8</sub>	N <sub>y9</sub>	.....	N <sub>ym</sub>

Where N is the node number, m is the number of nodes, x is the x-coordinate and y is the y-coordinate. The dimension of each particle is 2 × m (2-rows, m -columns) or 1 × (2m). For a population size of 10, a (10 × 2m) vector was formed.

**Objective Function Representation**

In order to evaluate the fitness of each particle, the fitness function that was used is given as;

$$F(Nx, y) = \frac{(\alpha \times \beta) \times 8}{T} \quad 1$$

Where, F (N<sub>x,y</sub>) is the fitness function of N<sub>x,y</sub> particle, α is packet through, β is the payload size and T is the total time taken.

**Performance Evaluation Result**

Five experiments were carried out for different population size (2, 4, 6, 8, and 10) of particles and for each, the effect of increasing population size was recorded and the average

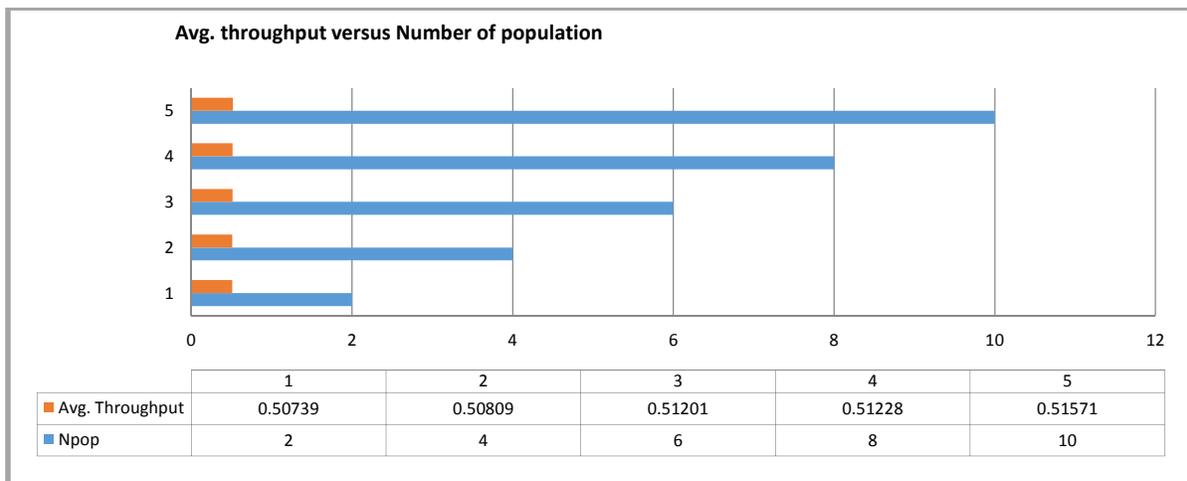
throughput for different number of population was computed as shown in Table 2. The graph of throughput versus number of population is shown in Figure 5.

The scalability of the system was also investigated for different network sizes (3, 6,

9, 12, 16), and for each the effect of network size on throughput was recorded and the average throughput for different number of nodes was computed in Table 3 and the graph of average throughput versus number of nodes is shown in Figure 6.

**Table 3** Average throughput for different number of population

S/No	Number of populations	Average Throughput
1	2	0.50739
2	4	0.50809
3	6	0.51201
4	8	0.51228
5	10	0.51571

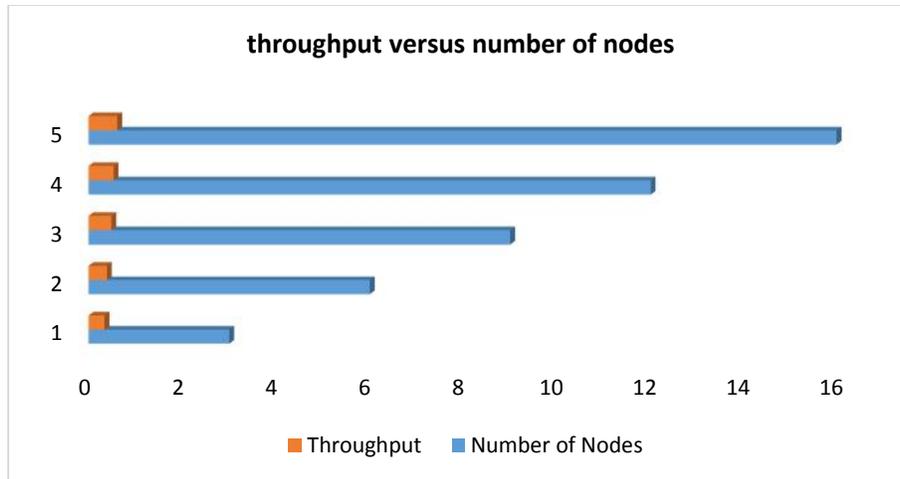


**Figure 5** Avg. throughput versus Number of population

### SCALABILITY INVESTIGATION RESULT

**Table 4** Average Throughput for different number of nodes

S/No.	Number of Nodes	Avg. Throughput
1	3	0.34707
2	6	0.38969
3	9	0.48979
4	12	0.54028
5	16	0.60600



**Figure 6** throughput versus number of nodes

## RESULTS AND DISCUSSION

The simulation results clearly show the following:

- i. Optimizing the sensor node placement with PSO algorithms yielded fast convergence and better solutions.
- ii. The network throughput also increases as the number of population increases, hence better result.
- iii. The throughput also increases as the number of nodes increase, hence the system support scalability.
- iv. The overall system performance was good based on realization of the objective function using PSO and dijkstra algorithms.

## CONCLUSION

This research work focused on optimization of wireless sensor nodes placement using particle swarm optimization algorithm for fire/event detection. PSO algorithm was used to optimized sensor nodes placement and in the end yielded fast convergence rate with better solution. The use of PSO algorithm was considered particularly for its fast and efficient performance, use of few parameters and fast convergence. Dijkstra shortest path algorithm was used in routing data through

the shortest path to the sink. The two algorithms used for the simulation help in the realization of the objective function.

## RECOMMENDATIONS

In future works, some specific recommendations are provided as follow:

- i. The system should be used in early fire detection and other natural disasters since it was modeled for such.
- ii. The scope of the system can be extended to support big organizations and industries since it supports scalability.
- iii. Other possible optimization techniques could be considered in future works to enhance system performance.

## REFERENCES

- [1] C. Fischione, "An Introduction to Wireless Sensor Networks," no. September, 2014.
- [2] Umashankar M L, and Ramakrishna M.V. "Optimization Techniques in Wireless Sensor Networks : A Survey," vol. 6, no. 7, pp. 704-708, 2017.
- [3] Mirjana Maksimovic and Vladimir Milosevic, "EVALUATING THE OPTIMAL SENSOR," no. January 2015,

- 2018.
- [4] Takawira J. Mumanga *et al.*, “Smart Emergency Response Interface for Fire Detection,” pp. 1–4, 2006.
- [5] F. Saeed, A. Paul, A. Rehman, W. H. Hong, and H. Seo, “IoT-Based Intelligent Modeling of Smart Home Environment for Fire Prevention and Safety,” pp. 1–16, 2018.
- [6] H. Soliman and A. Mishra, “A Smart Forest-Fire Early Detection Sensory System: Another Approach of Utilizing Wireless Sensor and,” pp. 1900–1904, 2010.
- [7] G. Laneve, A. Member, M. M. Castronuovo, and E. G. Cadau, “Continuous Monitoring of Forest Fires in the Mediterranean Area Using MSG,” vol. 44, no. 10, pp. 2761–2768, 2006.
- [8] J. Vicente and P. Guillemant, “An image processing technique for automatically detecting forest fire,” vol. 41, pp. 1113–1120, 2002.
- [9] B. U. Toreyin and A. E. Cetin, “Wildfire detection using LMS based active learning WILDFIRE DETECTION USING LMS BASED ACTIVE LEARNING,” no. June 2014, 2009.
- [10] A. K. Singh, “Forest Fire Detection through Wireless Sensor Network using Type-2 Fuzzy System,” vol. 52, no. 9, pp. 19–23, 2012.
- [11] V. Khanna, “Fire Detection Mechanism using Fuzzy Logic,” vol. 65, no. 12, pp. 5–9, 2013.
- [12] M. Dener, Y. Özkök, and C. Bostanc, “Fire Detection Systems in Wireless Sensor Networks,” vol. 195, pp. 1846–1850, 2015.
- [13] J. Zhang and J. Kan, “Forest fire detection system based on a ZigBee wireless sensor network,” no. September, 2008.
- [14] k. murungesa and D Ramesu, “Distributed mobile sink routing for wireless sensor networks: a survey,” Communications Surveys & Tutorials, IEEE.” International Journal of Inovative Reasearch in Computer Science and Engineering(IJIRCSE), pp. 1- 6, 2016.
- [15] X. Xu, W. Liang, and T. Wark, “Data Quality Maximization in Sensor Networks With a Mobile Sink.”
- [16] F. Y. S. Lin, P. L. Chiu, and S. Member, “A Near-Optimal Sensor Placement Algorithm to Achieve Complete Coverage / Discrimination in Sensor Networks,” vol. 9, no. 1, pp. 43–45, 2005.
- [17] J. Aguilar, E. Camargo, and A. Rios, “An Approach for Diagnosability Analysis and Sensor Placement for Continuous Processes Based on Evolutionary Algorithms and Analytical Redundancy,” no. January, 2015.
- [18] G. Wang, J. Cao, and J. I. E. Wu, “Sensor Placement with Multiple Objectives for Structural,” vol. 10, no. 4, 2014.
- [19] W. Liao, Y. Kao, and R. Wu, “Expert Systems with Applications Ant colony optimization based sensor deployment protocol for wireless sensor networks,” *Expert Syst. Appl.*, vol. 38, no. 6, pp. 6599–6605, 2011.
- [20] S. Torbey and S. G. Akl, “Reliable Node Placement in Wireless Sensor Networks Using Cellular Automata,” pp. 210–211, 2012.
- [21] Q. Bai, “Analysis of Particle Swarm Optimization Algorithm,” vol. 3, no. 1, pp. 180–184, 1998.
- [22] H. Kaur and S. Kautish, “An Implementation of Wireless Sensor Network Using Voronoi \_ PSO ( Particle Swarm Optimization ),” vol. 4, no. Xi, pp. 361–368, 2016.
- [23] I. C. Trelea, “The particle swarm optimization algorithm : convergence analysis and parameter selection,” vol. 85, pp. 317–325, 2003.

