DOI 10.12694/scpe.v20i1.1464
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 20, Number 1, pp. 71–82. http://www.scpe.org © 2019 SCPE

# A PARTICLE SWARM OPTIMIZATION BASED LOAD SCHEDULING ALGORITHM IN CLOUD PLATFORM FOR WIRELESS SENSOR NETWORKS

ARVINDA KUSHWAHA[*]AND MOHD AMJAD[†]

**Abstract.** Integration of wireless sensor network into cloud computing is a growing paradigm that supports a massive amount of applications in cloud computing, optimization of resources required in the machines. This integration requires the optimization of resources to efficiently complete the different tasks in the devices at cloud platform. This optimization can be done using load scheduling algorithms. These algorithms reduce overload and achieve higher throughput by maximizing the machine utilization concerning cost stabilization. There are lots of methods like First Come First Serve, Min-Min, Particle Swarm Optimization (PSO) for optimizing the load but we use Particle Swarm Optimization as it obtains the motivation from the social behavior of the flock of birds and analyses various approaches for load scheduling. In this paper, we propose the load scheduling algorithm based on PSO in wireless sensor networks for cloud computing to minimize total transfer time and cost stabilization. The proposed method is compared with the existing approaches used for load scheduling in Cloudlets. It is clear from the simulation results that the proposed method is more efficient because it minimizes the transfer time and cost than the conventional algorithms thereby making a system for cost stable.

**Key words:** Wireless Sensor Networks, Particle Swarm Optimization, Load Scheduling, Cloudlets, Cloud Computing.

**AMS subject classifications.** 68Q25, 68R10, 68U05

**1. Introduction.** The Wireless Sensor Networks (WSNs) have seen significant growth in academia as well as the industry in the field of WSNs due to the advancement in the sensor's abilities like sensing power, computation, and communication capabilities, in last decade and so. WSNs possess different characteristics like energy constraints, fault tolerance, heterogeneity and homogeneity of nodes, deployment scalability, ease of use, and adaptability to sustain in extreme environmental conditions [1, 2]. In WSN, various sensors are spread in the target area for supervising and logging the physical conditions of the environment that have a small size, low processing power, less storage, and low energy abilities. These sensor nodes can sense the target and make an infrastructure-less wireless communication among them and base station (BS). Sensors accumulate the data from the target area and forward it to the BS directly or with the assistance of other sensors. The BS is connected to the cloud server with the help of wired/wireless links. Thus, the information is accessed by the users from the cloud servers [3]. The BS is supposed to be reliable and is capable of performing any operation. WSNs are being used in different applications like battlefield surveillance, data centre monitoring and data logging, health care supervision, forest fire detection, landslide detection, natural disaster prevention, water quality monitoring, structural health monitoring, environmental conditions such as extreme temperature variation, sound, pollution levels, humidity, wind, etc., industrial and consumer applications like monitoring of computer system health, industrial process supervision and control etc., and so on [4]. On the bases of the above-discussed applications, the wireless sensor node deployment can be categorized into two categories like deterministic and non-deterministic. In deterministic deployments, sensor nodes are placed into controlled manner or manually at the selected locations where the deployment area is physically accessible such as city sense monitoring, soil monitoring, etc. On the other hand, in non-deterministic deployments sensor nodes are deployed into physically inaccessible areas using other sources like sensors are dropped from an aircraft, e.g., battlefield surveillance and landslide detection, etc. The non-deterministic deployment is also called random deployment [5, 6].

Due to the widespread advent of the Internet of Things (IoT) which connects daily use objects such as mobile devices, Smart TVs, washing machines, Air conditioners, etc. to the internet so that an intelligent linking can be done. Therefore, the user of the devices can communicate with the tools as per his/her convenience being either at home or office. One of the essential parts of the Internet of Things paradigm is wireless sensor networks (WSNs). To expand the wireless services with the online user base, there is a need of efficient hosting of the aggregated data from WSN on the cloud which is very flexible and cost-effective solution, so that online user

[*]Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India, ID (arvindakush@gmail.com)
[†]Department of Computer Engineering, Jamia Millia Islamia, New Delhi, India, ID ((amjad2k8@gmail.com)

can intelligently and efficiently communicate [7].

So, in this paper, we introduce a cost-efficient architecture for hosting the WSN data on the cloud platform. The architecture uses PSO based scheduling algorithm for load balancing among the servers of cloud so that the cost and time can be minimized.

This paper is framed in the following manner. The literature review of the integration techniques of wireless sensor networks and cloud computing is discussed in 2. A system model that includes the assumption for wireless sensor networks and cloud, network model is presented in 3. The proposed algorithmic method including flowchart is discussed in 4. The experimental setup and results are presented in 5. Finally,the paper provides the conclusions in 6.

**2. Literature Review.** In this section, we will discuss the literature review based on the integration of cloud platform and wireless sensor networks.

In paper [14], authors discuss an integrating framework for cloud paradigm and WSNs, which use data processability and cloud service model entirely. In the frame, data are efficiently utilized and managed to form the WSN by considering different information services provided to the users. In paper [15], an author discusses different types of scheduling algorithms and compares their results with various parameters. It also shows the many limitations of the existing algorithms such as they have not considered the execution time for calculating the performance of the method.

In paper [16], a trust assisted sensor cloud system is discussed which focuses on improving the quality of service of sensor cloud for users to obtain sensory data from the cloud. In faith supported sensor cloud system, trusted sensors which have trust values more significant than a threshold, collect and transmit sensory data to the cloud. Then the cloud selects the trusted data centers, which have trust values more significant than a threshold, to store, process the sensory data and further transmit the processed sensory data to users on demand. They show the trust assisted sensor cloud system can substantially improve the response time for users to obtain sensory data from the cloud system and compared to without faith supported sensor cloud system. In paper [17], the authors discussed wireless sensor network packet simulator, server application and web service, developed at the Institute Mihajlo Pupin for storing and viewing data collected by the network. The paper [18] discusses a novel architecture based on cloud computing for improving the performance of WSNs. In this architecture, a cloud platform acts as a virtual sink with the base station that collects data from sensors and processed in the distributed manner in the cloud system.

The paper [19] discusses an extensible and flexible architecture for integrating WSNs with the Cloud called REST. The REST-based Web services are used as an interoperable application layer that can be directly integrated into other application domains for remote monitoring. The paper [20] discusses a data processing framework for transmitting desired data to the mobile users. It decreases the storage requirements for sensor nodes and networks gateway and minimizes the traffic overhead and bandwidth requirement. However, the framework can able to predict the future trends of the sensory data and provides security for this sensory data.

The paper [23] analyzes the characteristics of job scheduling concerning integration cloud computing and wireless sensor networks and then discusses two popular job scheduling algorithms namely Min-Min and Max-Min. Two scheduling algorithms are proposed with the integration of cloud computing and wireless sensor networks by considering the Min-Min, and Max-Min and results of the proposed algorithms show shorter expected completion time than Min-Min and Max-Min, for cloud computing integrated with WSN. The paper [24] shows the optimal computing node searched in a limited time by proposing an ant colony algorithm. Three timers were used in setting back time for three role ants such as information and, positioning ant and task ant. When timer started, the general node of WSN was in power saving state. It takes a shorter time for the optimal computing resource. The paper [25] discusses location sleep scheduling algorithm with load balancing (LCLSS) scheme. It considers both awake and asleep status of sensors dynamically. It performs scheduling purely by mobile users current location. It reduces the power consumption and applies to remove areas applications where replacement of the battery is quite difficult. In the next section, we will discuss the system model and the assumptions for the integration of cloud computing and wireless sensor networks.

**3. System Model.** In this section, we discuss our system model by considering the following assumptions for the cloud networks:

- All nodes have similar capabilities, energy heterogeneity, location unaware, memory and computation constraints; a unique ID recognizes each node.
- The base station is stationarily located at the center in the network, and the sensor battery recharge or replacement is not possible.
- Each node has the capability to aggregate data, and the received signal strength can calculate the distance between nodes.
- The radio link consumes same energy in transmission and receiving information thus these links called symmetric connections.
- WSNs transmit gathered data to the cloud and end users require sensory data from the cloud.
- Multiple data centers are considered which consist of numerous data server in the cloud.

This network model consists of 3-types of classes based on their initial energy. In this network, WSN has n nodes out of which $\phi * n$ nodes have minimum energy, where a range of $\phi$ is $0 \leq \phi \geq 1$. We call these nodes Class-1 nodes and their energy is denoted as $E_1$. The $\phi^2 * n$ nodes have more energy than the Class-1 nodes, call these nodes Class-2 nodes and their energy is denoted as $E_2$. The remaining nodes have maximum energy called Class-3 nodes, i.e., (n-($\phi$*n+$\phi^2$*n)) and $E_3$ denotes their energy. The sensor nodes have maximum energy considered to be minimum in numbers. The total energy of the network ($Total_{energy}$) is given below:

$$Total_{energy} = \phi * n * E_1 + \phi^2 * n * E_2 + (1 - \phi - \phi^2) * n * E_3 \tag{3.1}$$

This model describes a WSN, i.e., consisting of Class-1, Class-2, and Class-3 heterogeneity by considering the value of model parameter $\phi$. The range of $\phi$ is between 0 and 1. When we put $\phi$=0 in (3.1), it gives one non-zero term which contains only Class-3 nodes. It indicates the 1-level of heterogeneity, but these nodes have $E_3$ energy. In the case of 1-level heterogeneity, appropriate constraints are imposed to haveClass-1 nodes instead of the Class-3 nodes. It may be calculated by defining the following equation:

$$\phi = \frac{E_3 - E_1}{N * f(E_2, E_3)} \tag{3.2}$$

where $N$ and $f$ are the positive integer greater than 1 and the function of $E_2$ and $E_3$, respectively. The function has either ($E_3$+$E_2$) or ($E_3$-$E_2$). In (3.2), the value of $\phi$ should be in the consonant by considering the constraint: $E_1 < E_2 < E_3$.

The 2-level of heterogeneity contains two type of nodes, i.e., Class-1 and Class-2 for this we need to find the value of $\phi$ by considering the following equation:

$$1 - \phi - \phi^2 = 0 \tag{3.3}$$

Equation (3.3) is generated by (3.1) by considering the third term which generates the two level of heterogeneity. We call these nodes Class-1 and Class-2 nodes. Equation (3.3) has two solutions: (($\sqrt{5}$)-1)/2 and (($\sqrt{5}$)+1)/2. Since $\phi$ is upper-bounded by 1 and (($\sqrt{5}$)+1)/2>1, the valid solution of (3) is (($\sqrt{5}$)-1)/2. For $\phi$ =(($\sqrt{5}$)-1)/2, the model (1) consists of two types of nodes with energies $E_1$ and $E_2$.

In the case of 3-level heterogeneity, the range of $\phi$ is firstly determined where the upper limit is (($\sqrt{5}$)-1)/2. Let the lower limit of $\phi$ be $\phi_L$ that is to find out. The range of $\phi$ for 3-level heterogeneity is $\phi_L$<$\phi$<(($\sqrt{5}$)-1)/2. Taking $f$ as ($E_3$-$E_2$) and $\phi$ from (3.2), we have

$$\phi_L < \frac{E_3 - E_1}{N * (E_3 - E_2)} < ((\sqrt{5}) - 1)/2 \tag{3.4}$$

Let $E_2$=$\alpha_1$ +$E_1$ and $E_3$=$\alpha_2$+$E_2$. From (3.4), we have

$$\frac{\alpha_2}{\alpha_1} < \frac{1}{n * \theta_L - 1} Or \frac{-\alpha_2}{\alpha_1} \geq \frac{1}{1 - n * \theta_L} \tag{3.5}$$

Since L.H.S. of inequality (5) is negative, we should have 1-N*$\phi_L$ < 0. This gives

$$\frac{1}{N} < \phi_L \tag{3.6}$$

The (3.4) can be written as

$$(E_3 - E_1) \leq N * \frac{(\sqrt{5}) - 1}{2} * (E_3 - E_2) \tag{3.7}$$

This inequality may be written as

$$N * ((\sqrt{5}) - 1) * E_2 - 2 * E_1 \leq (N * ((\sqrt{5}) - 1) - 2) * E_3 \tag{3.8}$$

This model describes WSNs which consists of three types of nodes, i.e., Class-1, Class-2, and Class-3. The total energy of the network (1) also describes the 1-level, 2-level, and 3-level of heterogeneity. In the next section, we will discuss the proposed clustering method for the heterogeneous network model.

**4. Proposed Method.** In this section, we discuss the proposed method which is divided into two parts namely energy efficient clustering protocols for WSNs and load scheduling algorithm based on PSO for cloud computing.

**4.1. Energy Efficient Clustering Protocol For Sensor Networks.** In this section, we consider an energy efficient heterogeneous DEEC protocol in WSNs [1]. The 3-level heterogeneous network model consists of three types of sensor nodes which are Class-1, Class-2, and hetDEEC-1, hetDEEC-2 denote Class-3 and its implementation, and hetDEEC-3, respectively. The cluster head selection of [1] assumes N* $p_{opt}$ as the average number of cluster heads as in every iteration, and each sensor node assumes the responsibility of a cluster head once in every $r_i$=1/$p_{opt}$ iteration, where $p_{opt}$ is an initial probability of each sensor node, and $r_i$ is iteration. The networks average energy E (r)can be calculated as follows:

$$E(r) = \frac{1}{N}\sum_{i=1}^{N} E_i(r) \tag{4.1}$$

where $E_i(r)$ is residual energy. The average probability of $i^{th}$ node for becoming the cluster head during $r^{th}$ iteration can be calculated as follows:

$$p_i = p_{opt}[1 - \frac{E(r) - E_i(r)}{E(r)}] = p_{opt} * \frac{E_i(r)}{E(r)} \tag{4.2}$$

The count of cluster heads per iteration is given by

$$\sum_{i=1}^{N} p_i = \sum_{i=1}^{N} p_{opt} * \frac{E_i(r)}{E(r)} = p_{opt}\sum_{i=1}^{N}\frac{E_i(r)}{E(r)} = N * p_{opt} \tag{4.3}$$

The cluster head in $r^{th}$ iteration for the $i^{th}$ node is given by (using (4.2))

$$r_i = \frac{1}{p_i} = \frac{E(r)}{p_{opt} * E_i(r)} = r_{opt} * \frac{E(r)}{E_i(r)} \tag{4.4}$$

As discussed in (3.1), the total energy at the beginning of the sensor network is $N*(\phi^*E_1+\phi^{2*}E_2+(1\text{-}\phi\text{-}\phi^2)^*E_3)$ which is increased by $\phi+\phi^{2*}E_2/E_1$ +(1-$\phi$-$\phi^2$ )*$E_3/E_1$. Here, all nodes assume the responsibility of cluster head exactly once in every $\frac{1}{p_{opt}}$ *($\phi+\phi^{2*}E_2/E_1$ +(1-$\phi$-$\phi^2$ )*$E_3/E_1$ )iteration. Thus, the average number of cluster heads per iteration is($\phi+\phi^{2*}E_2/E_1$ +(1-$\phi$-$\phi^2$ )*$E_3/E_1$ )*N*$p_{class-1}$. In1-level of heterogeneity, each Class-1 node becomes a cluster head once in every ($\phi+\phi^{2*}E_2/E_1$ +(1-$\phi$-$\phi^2$ )*$E_3/E_1$) iteration, each Class-2 node becomes a cluster head (1+$\alpha$) times more than the Class-1 nodes in every (($\phi+\phi^{2*}\frac{E_2}{E_1}$ +(1-$\phi$-$\phi^2$ )*$\frac{E_3}{E_1}$) iteration, and each Class-3 node becomes a cluster head (1+$\beta$) times more than the Class-1 nodes in every (($\phi+\phi^{2*}E_2/E_1$ +(1-$\phi$-$\phi^2$ )*$E_3/E_1$ )) iteration. These methods mainly associate weights to obtain the optimal probability for each class of sensor node. It divides $p_i$ by the factor of the total increased energy in the network for clustering

[1]. The weighted probabilities of the Class-1, Class-2, and Class-3 nodes denoted by $p_{class-1}$, $p_{class-2}$, and $p_{class-3}$, respectively, are given by

$$p_{class-1} = \frac{p_{opt} * E_i(r)}{(\phi + \phi^2 * E_2/E_1 + (1 - \phi - \phi^2) * E_3/E_1) * E(r)} \tag{4.5}$$

$$p_{class-2} = \frac{p_{opt} * (1 + \alpha) * E_i(r)}{(\phi + \phi^2 * E_2/E_1 + (1 - \phi - \phi^2) * E_3/E_1) * E(r)} \tag{4.6}$$

$$p_{class-3} = \frac{p_{opt} * (1 + \beta) * E_i(r)}{(\phi + \phi^2 * E_2/E_1 + (1 - \phi - \phi^2) * E_3/E_1) * E(r)} \tag{4.7}$$

The cluster head selection probability is calculated as follows.

$$\begin{aligned} T(s) &= \frac{p_{opt}}{1 - p_{opt}.(rmod * 1/p_{opt})} \quad ifs \in G \\ &= 0 \ otherwise \end{aligned} \tag{4.8a}$$

The thresholds T($s_i$) Class-1, Class-2, and Class-3 nodes are given by

$$\begin{aligned} T(s_i) &= \frac{p_{class-1}}{1 - p_{class-1} * (rmod * 1/p_{class-1})} \quad ifp_{nrm} \in G' \\ &= \frac{p_{class-2}}{1 - p_{class-2} * (rmod * 1/p_{class-2})} \quad ifp_{adv} \in G'' \\ &= \frac{p_{class-3}}{1 - p_{class-3} * (rmod * 1/p_{class-3})} \quad ifp_{sup} \in G''' \\ &= 0 \ otherwise \end{aligned} \tag{4.9a}$$

where $G'$, $G''$ and $G'''$ are set of Class-1, Class-2, and Class-3 nodes that have not become cluster heads within last $\frac{1}{p_{Class-1}}$ ,$\frac{1}{p_{Class-2}}$ ,and $\frac{1}{p_{Class-3}}$ rounds, respectively.

**4.2. Load Scheduling Algorithm Based On Particle Swarm Optimization For Cloud Computing.** In this subsection, we discuss the proposed scheme for load scheduling which is based on simple and basic particle swarm optimization [8, 9]. A new fitness function is incorporating in the proposed method. The proposed scheme defines the fitness function by the transfer time and costs with the higher exploitation of the space of the particles in the base station (BS). The data packets are initialized to the cloudlets and the virtual machines from the BS. The cloudlets are passed to the particles ($N_p$) to be allocated to the optimal virtual machines. The Cost(T) is specified as the total cost of all the tasks are assigned to calculate the resources available. The Cost(T) is calculated as follows:

$$C_{ex}(T)_j = \sum_k W_{kj} \tag{4.10}$$

$$\forall T(k) = j \tag{4.11}$$

$$C_{total}(T)_j = C_{ex}(T)_j + C_{tr}(T)_j \tag{4.12}$$

$$Cost(T) = \max(C_{total}(T)_j) \ \forall j \in P \tag{4.13}$$

$$\min(Cost(T) \ \forall T) \tag{4.14}$$

Here $C_{ex}(T)_j$, $C_{tr}(T)_j$, and $C_{total}(T)_j$ are the execution time, the transfer time employed in passing the particles from one resource to another, and gives the sum of the execution cost and the transfer cost between the tasks and the resources for the particle j, respectively. The transfer time is the maximum time taken by all the VMs to complete a particular task. The transfer time is calculated as:

$$E_{vmtime}(T)_k = \sum_k W_k \tag{4.15}$$

$$TransTime(T) = \max\left(E_{vmtime}(T)_k\right) \forall j \in N \tag{4.16}$$

$$k = 1, .., N \tag{4.17}$$

The new fitness function for both the cost and transfer time parameters is minimized by considering the weights in the system. The new fitness function is calculated as follows.

$$\alpha * (Cost(T)) + (1 - \alpha) * TransTime(T) \tag{4.18}$$

$$\min(\alpha(Cost(T)) + (1 - \alpha)TransTime(T)) \tag{4.19}$$

where $\alpha$ is the random parameter range from 0 to 1. It is used to shift the load scheduler to both the parameters, i.e., cost and transfer time giving the weightage to both the parameters of the packets in the base station.

The proposed fitness function minimizes weighted sum of transfer time and cost. If the value of $\alpha < 0.5$, higher weightage to the cost function otherwise, higher weightage to the transfer time. The ring-shaped clusters are employed in the neighborhood of the particle to obtain a new global best. The cumulative best values of the clusters help in moving the particles further to the next position.

These motivate in finding the best particles among all the particles, i.e., pbest(i,t) and global best among all the particles in the current iteration, i.e., gbest(t) values.The pbest value is computed as follows:

$$pbest(i,t) = arg \min_{k=1,....t} [f(P_i(k))], i \in 1, .....N_p \tag{4.20}$$

and gbest is known as the best position:

$$gbest(t) = arg \min_{i=1,....N_p; k=1,....t} [f(P_i(k))] \tag{4.21}$$

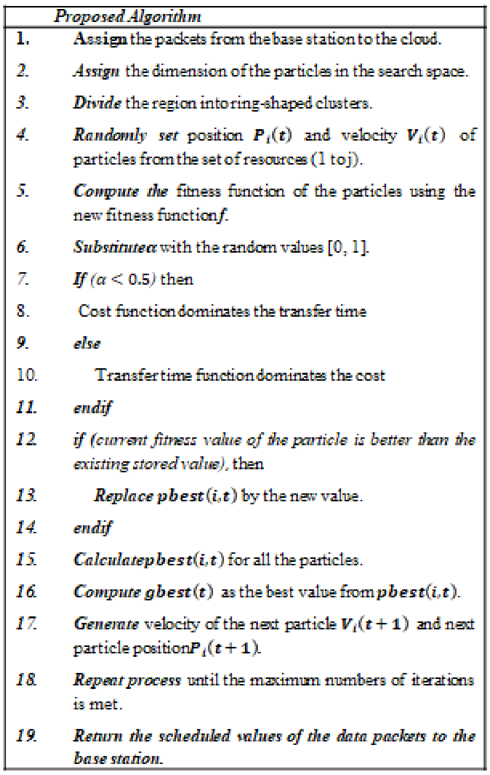
Here, $i, N_p, f, P, t$ represent the index of the particle, a total number of particles, symbolically-represent the fitness function, position, and current round/iteration, respectively. The velocity and position of the particles are computed in the following manner.

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pbest(i,t) - P_i(t)) + c_2 r_2 (gbest(t) - P_i(t)) \tag{4.22}$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \tag{4.23}$$

where the velocity of the particle i at iteration t is denoted by $V_i(t)$, the velocity of the particle i at (t+1 )round is denoted as $V_i(t+1)$. The $c_1$ and $c_2$ represent the acceleration coefficients of the system & $r_1$ and $r_2$ denote the random values in the range 0 to 1with $\omega$ denoting the inertia weight. The best location of the particle i is denoted by pbest(i,t) and the best position among all pbest(i,t) values is generated by gbest(t). The $P_i(t)$ specifies the current position of the particle i at iteration t and $P_i(t+1)$ denotes the position of the particle i at (t+1) iteration.On the basis of these values, the particles migrate to the next location $P_i(t+1)$. The positions of the VMs assigned to the particles are returned for further execution to the cloudlets in the cloud over the datacenter which further schedule the packets in the base station. These values are returned back to the base station.

The complete flow diagram of the proposed method is shown in Fig. 4.2.



```
Proposed Algorithm
Assign the packets from the base station to the cloud.
Assign the dimension of the particles in the search space.
Divide the region into ring-shaped clusters.
Randomly set position $P_i(t)$ and velocity $V_i(t)$ of particles from the set of resources (1 toj).
Compute the fitness function of the particles using the new fitness function $f$.
Substitute $\alpha$ with the random values [0, 1].
If ($\alpha < 0.5$) then
    Cost function dominates the transfer time
else
    Transfer time function dominates the cost
endif
if (current fitness value of the particle is better than the existing stored value), then
    Replace $pbest(i,t)$ by the new value.
endif
Calculate $pbest(i,t)$ for all the particles.
Compute $gbest(t)$ as the best value from $pbest(i,t)$.
Generate velocity of the next particle $V_i(t+1)$ and next particle position $P_i(t+1)$.
Repeat process until the maximum numbers of iterations is met.
Return the scheduled values of the data packets to the base station.
```

Fig. 4.1. *Proposed Algorithm.*

Table 5.1
*Simulation parameters for the radio dissipation model and network model for wireless sensor networks [1,3,4]*

| Description | Value |
|---|---|
| Battery consumed to transmit at a shorter distance | 10nJ/bit/m2 |
| Battery consumed to transmit at a longer distance | 0.0013pJ/bit/m4 |
| Battery consumed to transmit or receive the signal | 50nJ/bit |
| The battery used in data aggregation | 5nJ/bit/signal |
| Threshold distance | 70 m |
| Message Size | 4000 bits |
| Network Size | 100M X 100M |
| Base station Position | (50,50) |
| Maximum no. of Sensor Nodes | 100 |
| Cluster Radius | 25M |
| Initial battery | 0.50J |
| Constant N | 10 |
| 1-level of heterogeneity: no of nodes and initial battery | 100 and 0.5J |
| 2-level of heterogeneity: no of nodes and initial battery of nodes | 80 & 20 and 0.5J & 1.5J |
| 3-level of heterogeneity: no of nodes and initial battery of nodes | 51, 26, & 23 and 0.5J, 1.38J & 1.67J |

**5. Experimental Results And Discussions.** The simulation considers the random deployment of 100 wireless sensor nodes in a field of size 100M X100M. It locates the base station at the center of the area, and we have borrowed a radio dissipation model from [1, 3, 4]. The model and associated input parameters are given in Table 5.1.

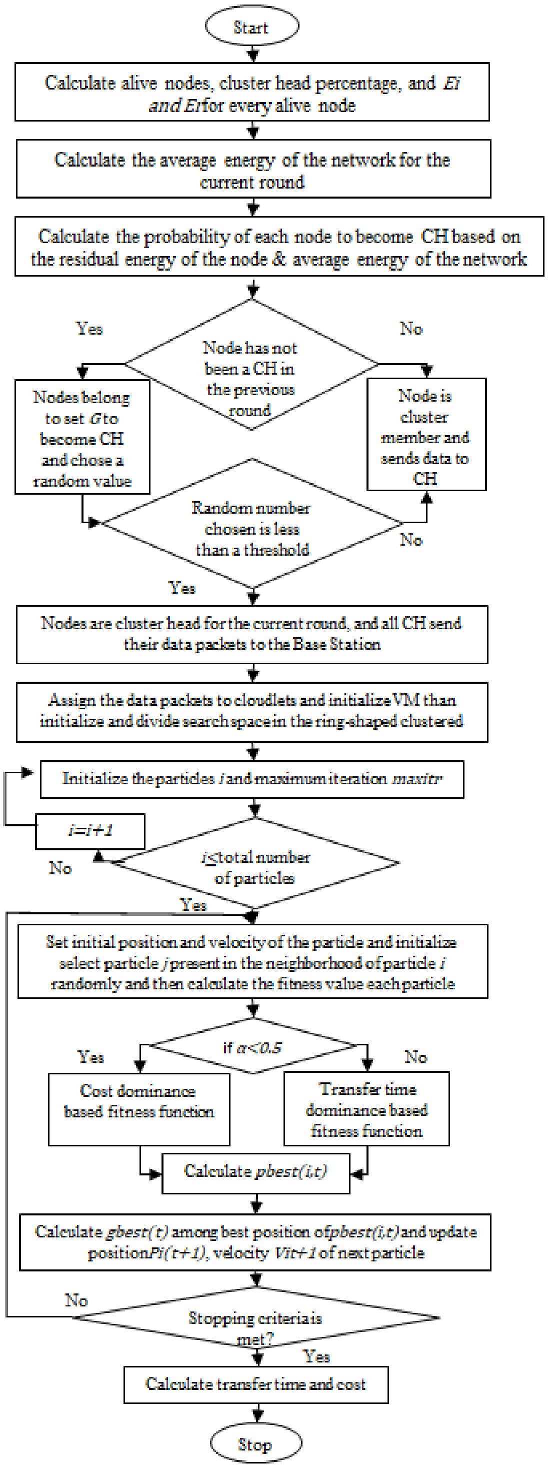

Start
Calculate alive nodes, cluster head percentage, and Ei and Erfor every alive node
Calculate the average energy of the network for the current round
Calculate the probability of each node to become CH based on the residual energy of the node & average energy of the network
Yes
No
Node has not been a CH in the previous round
Nodes belong to set G to become CH and chose a random value
Node is cluster member and sends data to CH
Random number chosen is less than a threshold
No
Yes
Nodes are cluster head for the current round, and all CH send their data packets to the Base Station
Assign the data packets to cloudlets and initialize VM than initialize and divide search space in the ring-shaped clustered
Initialize the particles i and maximum iteration maxitr
i=i+1
No
i≤total number of particles
Yes
Set initial position and velocity of the particle and initialize select particle j present in the neighborhood of particle i randomly and then calculate the fitness value each particle
if α<0.5
Yes
No
Cost dominance based fitness function
Transfer time dominance based fitness function
Calculate pbest(i,t)
Calculate gbest(t) among best position ofpbest(i,t) and update positionPi(t+1), velocity Vit+1 of next particle
No
Stopping criteria is met?
Yes
Calculate transfer time and cost
Stop

FIG. 4.2. *Flowchart of the proposed method.*

The proposed method and existing PSO algorithm [8, 9, 10, 11] are implemented in the CloudSim tool package. The CloudSim is used to perform, designing and analyzing the results and provides users built-in environment. It was given by Garg et al. in 2011, and Network CloudSim Simulator is simulated on top of the CloudSim tool package [7]. We consider two important matrices namely transfer time (seconds) and cost (dollars) for comparing the results of the proposed and existing approaches. The proposed method uses a JSwarm package to perform the simulation. The work in consideration takes ten particles referring to VMs. Each particle's resources include the following parameters like inertia, maximum & minimum position, and velocity. The dynamic allocation of resources depends on the number of iterations. The iterations taken into consideration are 10, 50, 100, 200, 500, and 1000 in figures. The values are provided before running to the system. The cloudlets and VMs possess the same capability given by the system such as MIPS, transfer cost, execution cost (which are used for the computation of the execution cost and transfer cost), and bandwidth. The parameters supplied to the simulator or the workload characteristics are MIPS (1000), ram (2048), bandwidth (10000), storage (10000) along with the number of iterations.

The results of the proposed method and existing approaches are analyzed by considering two matrices, i.e., transfer time and cost and compared. Six values of the proposed method are considered for the categorization namely, minimum value ($\alpha = 0$), average value ($\alpha = 0.3$), best value ($\alpha = 0.4$), mid or half value ($\alpha = 0.5$), random value ($\alpha$= 0.7) and maximum value ($\alpha = 1$) values. These results are shown for a large set of iterations regarding transfer time and cost concerning the number of iterations.

The results are analyzed graphically as given in Fig. 5.1 to 5.4. Fig. 5.1 provides a precise depiction of the transfer time of the proposed approach at various $\alpha$ values, viz., $\alpha = 0$, $\alpha = 0.3$, $\alpha = 0.4$ (best value), respectively with existing ones PSO. Fig. 5.2 shows the current approaches with proposed method along with the various interval values regarding the total transfer time incurred over the number of iterations at $\alpha = 0.5$ (average value), $\alpha = 0.7$, $\alpha = 1$. Finally, it balances the system and decreases the cost of computation as well as the finish time.

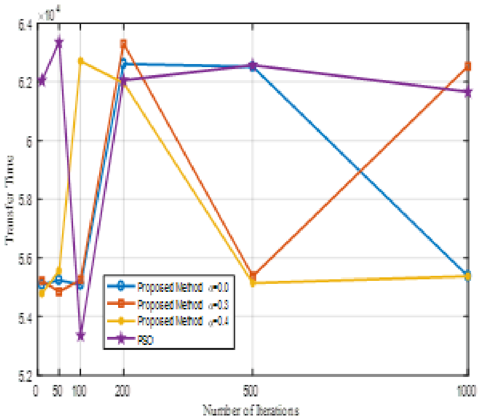


Fig. 5.1. *Shows the transfer time concerning the number of iteration for the existing PSO and proposed a method by considering the different value of α=0.0,0.3,and 0.4.*

Fig. 5.3 provides a representation of the cost of the proposed method concerning many iterations at different $\alpha$ values, viz., $\alpha = 0$, $\alpha = 0.3$, $\alpha = 0.4$ (best value), respectively with existing ones PSO. Fig. 5.4 illustrates cost over the number of iterations at $\alpha = 0.5$ (Average Value), $\alpha = 0.7$, $\alpha = 1$. The cost analysis provides much better results for a large number of iterations. The average cost at all the iterations is quite similar. The proposed method gives a stable view of the cost incurred in the system thereby creating a stabilized system. Thus, the proposed approach produces better results regarding transfer time with a stable cost, i.e., no increase in cost. Fig. 5.3 and 5.4 give the total cost differentiation for a larger value of iterations showing maximum optimization. It explains that cost differentiation generates optimal results at various values and best result at $\alpha = 0.4$ (Best Value) and $\alpha = 0.5$ (Average Value). It shows that less cost is incurred in the proposed method than existing approaches.

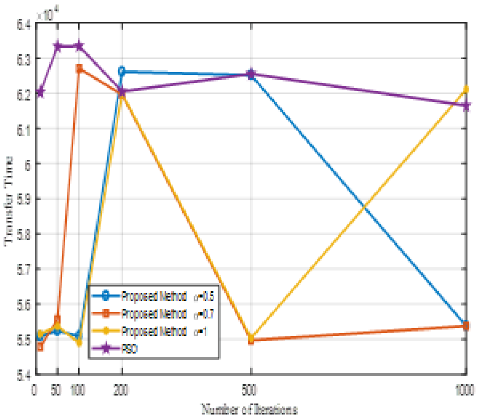


FIG. 5.2. *Shows the transfer time concerning the number of iteration for the existing PSO and proposed the method by considering the different value of α=0.5,0.7,and 1*

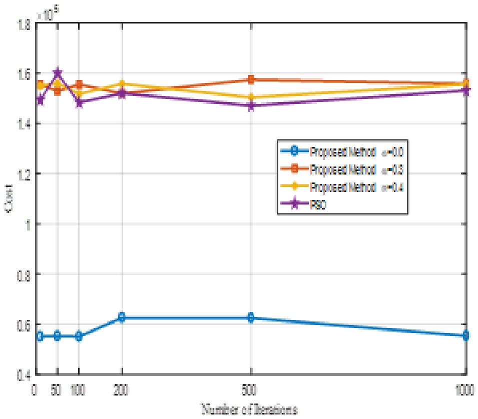


FIG. 5.3. *Shows the cost for the number of iteration for the existing PSO and proposed a method by considering the different value of* $\alpha = 0.0, 0.3$ *and 0.4*

It is evident from the Figs. 5.1 to 5.3 that the proposed method takes less transfer time concerning many iteration in the computation of the tasks. It leads to a further reduction in the total computational cost of the system. It produces much higher exploitation of the resources as compared to the existing approaches as evident from the figure. The figure signify the reduction in the total transfer time and total cost based on the proposed method. The cost and transfer time are the prime metrics to justify the storage, processing, and retrieval of data. Lesser transfer time leads to faster processing of user requests. Less cost specifies higher bandwidth and ram for more data processing. This proposed method provides more considerable significance in the real world scenario by optimal allocation, scheduling, and execution of the user requests on the virtual machines. Based on this the total cost incurred in the storage and processing of the data or information in the cloud is further reduced considerably by more considerable search space exploitation.

Table 5.2 shows the network lifetime regarding rounds by taking an equal number of nodes (i.e., 100) and the same amount of total network energy (i.e., 100J) 1-level, 2-level, 3-level heterogeneity. Categorization of nodes and their respective energies are given in details in Table 5.1. The number of rounds for 1-level (Class-1), 2-level (Class-2), 3-level (Class-3) heterogeneity are 2961, 3323, and 4404 in hours, respectively.

**6. Conclusions.** This paper discusses a load scheduling algorithm based on particle swarm optimization (PSO) in wireless sensor networks for cloud platform among the tasks. The primary objective of this method is to minimize transfer time and decrease the system cost. The total transfer time and the total cost are evaluated

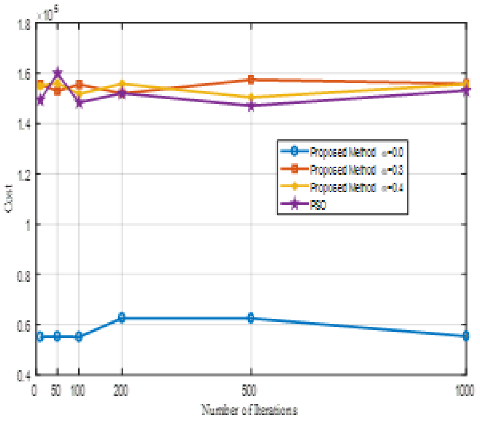


Fig. 5.4. *Shows the transfer time concerning the number of iteration for the existing PSO and proposed a method by considering the different value of $\alpha = 0.5, 0.7$ and 1*

Table 5.2
*Number of rounds for Class-1, Class-2, Class-3 using 100 number of nodes and 100 J network energy for 1-level, 2-level, and 3-level heterogeneity.*

| Classes | Nature of Networks | No. of Rounds (hrs) |
|---|---|---|
| Class-1 | 1-level heterogeneity | 2961 |
| Class-2 | 2-level heterogeneity | 3323 |
| Class-3 | 3-level heterogeneity | 4404 |

by the computational values viz. the fitness value of the method. The proposed method produces better results and also provides additional benefits by making the system more stable regarding cost. The proposed method achieves higher cost optimization than the existing contemporary PSO based approaching terms of transfer time and thereby making the system more stable on the cost basis. The transfer time and the cost are reduced comparatively at $\alpha = 0.4$ (Best Value) as shown in the Figures. The future work aims at minimizing the price further to obtain greater feasibility of the results in a cloud computing environment.

## REFERENCES

[1] S. Singh and A. Malik "hetDEEC: Heterogeneous DEEC Protocol for Prolonging Lifetime in Wireless Sensor Networks," Journal of Information and Optimization Sciences, vol. 38, no. 5, pp. 699-720, 2017.

[2] S. Singh, Energy Efficient Multilevel Network Model for Heterogeneous WSNs Int. Journal Engineering Science and Technology, vol. 20, no. 1, pp. 105-115, 2017.

[3] A. Malik and S. Singh "hetSEP: Heterogeneous SEP Protocol for Increasing Lifetime in WSNs," Journal of Information and Optimization Sciences, vol. 38, no. 5, pp. 721-743, 2017.

[4] S. Singh and A. Malik, Energy Efficient Scheduling Protocols for Heterogeneous WSNs International Journal of Forensic Computer Science, vol. 11, no. 1, pp. 8-29, 2016.

[5] A. Malik and S.Singh, "HeterogeneousEnergy Efficient Protocol for Enhancing the Lifetime in WSNs I.J. Information Technology and Computer Science, vol.8, no.9, pp. 62-72, 2016.

[6] S. Chand, S. Singh, and B. Kumar, "Multilevel Heterogeneous Network Model for Wireless Sensor Networks," Telecommunication Systems, vol. 64(2), pp. 259277, 2016.

[7] S.K. Garg, and R. Buyya, "Network CloudSim: Modelling Parallel Applications in Cloud Simulations," 4th IEEE/ACM International Conference on Utility and Cloud Computing (UCC 2011, IEEE CS Press, USA), Melbourne, Australia, 2011.

[8] J Kennedy and R Eberhart, Particle swarm optimization. IEEE International Conference on Neural Networks, 4, 19421948, 1995.

[9] S. Pandey, and R Buyya et al., A Particle Swarm Optimization based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments.24th IEEE International Conference on Advanced Information Networking and Applications, 400-407, 2010.

[10] R.Buyya, S Pandey and Vecchiola, Cloudbus toolkit for market-oriented cloud computing.CloudCom 09: Proceedings of the 1st International Conference on Cloud Computing, 2009.

[11] P.Y. Yin, S.S. Yu, and Y.T. Wang, "A hybrid particle swarm optimization algorithm for optimal task assignment in distributed systems" Computer Standards and Interfaces, 28(4), 441450, 2006.

[12] S. Singh, S. Chand, and B. Kumar, "Performance Evaluation of Distributed Protocols Using Different Levels of Heterogeneity Models in Wireless Sensor Networks," Int. Journal of Computer Network and Information Security, 7(1), pp.38-45, 2015.

[13] A K Sharma, and S. Singh, Distributed Algorithms for Maximizing Lifetime of WSN with Heterogeneity and Adjustable Range for Different Deployment Strategies I. J. Information Technology and Computer Science, 5(8), pp.101-108, 2013.

[14] P. You; Y. Peng; and H. GaoProviding Information Services for Wireless Sensor Networks through Cloud Computing IEEE Asia-PacificServices Computing Conference (APSCC), pp. 362 - 364, 6-8 Dec. 2012

[15] A.VenumadhavA Survey of Various Workflow Scheduling Algorithms in Cloud Environment International Journal of Scientific and Research Publications, Vol. 3, Issue 10, Oct. 2013.

[16] C. Zhu, V. C. M. Leung, L. T. Yang, L. Shu, J. J. P. C. Rodrigues, X. Li, "Trust Assistance in Sensor-Cloud" IEEE INFOCOM 2015, pp. 342-347, 2015.

[17] L. Kraus, M. Starcevic, M. Oklobdija and . Stojkovic, "Integrated Software Tools for Support of Wireless Sensor Network Applications", 19th Telecommunications forum TELFOR 2011 Serbia, Belgrade, pp. 1289-1292, November 22-24, 2011.

[18] P. Zhang, Z. Yan, and H. Sun, "A Novel Architecture Based on Cloud Computing for Wireless Sensor Network" Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013), pp. 472-475, 2013.

[19] R. Piyare, S. Park, S. Y. Maeng, S. H. Park, S. C. Oh, Sang G. Choi, H. S. Choi, and S. R. Lee, "Integrating Wireless Sensor Network into Cloud Services for Real-time Data Collection" ICTC 2013, pp. 752-756, 2013.

[20] C.M Sukanya et al., " Integration of Wireless Sensor Networks and Mobile Cloud- a Survey" International Journal of Computer Science and Information Technologies, vol. 6, no. 1, 159-163, 2015.

[21] Y. Singh, S. Singh, and R. Kumar A Distributed Energy-Efficient Target Tracking Protocol for Three Level Heterogeneous Sensor Networks, International Journal of Computer Applications, 51(11), pp-31-36, Aug. 2012.

[22] A K Sharma, and S. Singh Distributed Energy-Efficient Algorithm for Wireless Sensor Networks, International Journal of Advanced Research in Computer Science, 2(3), pp-548-550, June 2011.

[23] C. Zhu, X. Li, V. C. M. Leung, X. Hu, and L. T. Yang, "Job Scheduling for Cloud Computing Integrated with Wireless Sensor Network," IEEE 6th International Conference on Cloud Computing Technology and Science, pp. 62-69, 2014

[24] H. Yuan, C. Li, and M. Du, "Resource Scheduling of Cloud Computing for Node of Wireless Sensor Network Based on Ant Colony Algorithm". Journal Information Technology, vol. 11, pp.1638-1643, 2012.

[25] R. Priya, "collaborative location-based sleep scheduling with load balancing in sensor-cloud," International Journal of Electrical and Electronics Research, vol. 5, Issue 2, pp: (63-72), 2017.

*Edited by:* Khaleel Ahmad
*Received:* Nov 25, 2018
*Accepted:* Feb 11, 2019