Comparative Study on Ant Colony Optimization (ACO) and K-Means Clustering Approaches for Jobs Scheduling and Energy Optimization Model in Internet of Things (IoT)

Sumit Kumar[1], Vijender Kumar Solanki[2]*, Saket Kumar Choudhary[3], Ali Selamat[4,5], Rubén González Crespo[6]

[1] Department of Information Technology, Gopal Narayan Singh University, Sasaram, Bihar (India)
[2] Department of Computer Science & Engineering, CMR Institute of Technology, Hyderabad, TS (India)
[3] Saket Kumar Choudhary, FCA, MRIIS, Faridabad, Haryana (India)
[4] Media and Games Center of Excellence (MagicX) Universiti Teknologi Malaysia & School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, Skudai, Johor (Malaysia)
[5] Malaysia Japan International Institute of Technology (MJIIT), Universiti Teknologi Malaysia Kuala Lumpur, Jalan Sultan Yahya Petra, Kuala Lumpur (Malaysia)
[6] Department of Computer Science and Technology, Universidad Internacional de La Rioja, Logroño (Spain)

Abstract

The concept of Internet of Things (IoT) was proposed by Professor Kevin Ashton of the Massachusetts Institute of Technology (MIT) in 1999. IoT is an environment that people understand in many different ways depending on their requirement, point of view and purpose. When transmitting data in IoT environment, distribution of network traffic fluctuates frequently. If links of the network or nodes fail randomly, then automatically new nodes get added frequently. Heavy network traffic affects the response time of all system and it consumes more energy continuously. Minimization the network traffic/ by finding the shortest path from source to destination minimizes the response time of all system and also reduces the energy consumption cost. The ant colony optimization (ACO) and K-Means clustering algorithms characteristics conform to the auto-activator and optimistic response mechanism of the shortest route searching from source to destination. In this article, ACO and K-Means clustering algorithms are studied to search the shortest route path from source to destination by optimizing the Quality of Service (QoS) constraints. Resources are assumed in the active and varied IoT network atmosphere for these two algorithms. This work includes the study and comparison between ant colony optimization (ACO) and K-Means algorithms to plan a response time aware scheduling model for IoT. It is proposed to divide the IoT environment into various areas and a various number of clusters depending on the types of networks. It is noticed that this model is more efficient for the suggested routing algorithm in terms of response time, point-to-point delay, throughput and overhead of control bits.

Keywords

I. Introduction

THE latest IoT applications depend on promotion of wireless sensor networks (WSNs) with expert of engineering. These IoT applications contain a large number of devices, connected with different requirements and technologies. Such kinds of IoT applications do the sensing and collection of data with transmission of data to the administrator nodes for other possible operations and even a cloud at the backdrop for data analytics. These processes require routing protocols for their completion. Routing protocols have two major challenges. The first challenge is to improve data transmission and scalability whereas the second challenge is to minimize energy consumption. In an IoT application, network nodes under different network topology collect different kind of data so that an IoT application produces an enormous amount of data. The heterogeneity in network topology restricts the TCP/IP to become the best policy for proper resource allocation to computing and routing [1]-[3], [27]-[29].

Owing to the above-mentioned challenges, different persons view IoT in different ways, based on their perception and requirements. A routing protocol includes the multiple job scheduling methodologies. These job scheduling methodologies are reported as either heuristic or metaheuristic-based approaches. Heuristic-based methodologies are comparatively more helpful when we look for a local optimum whereas metaheuristic methodologies further try to explore the solution

* Corresponding author.

E-mail address: spesinfo@yahoo.com

space to attain global optima. Despite the fact that the metaheuristic methodologies look very engaging and a large number of parameters to be turned on account of IoT thus limits the utilization [4]-[9], [27]-[29].

A number of researchers have developed and used the ACO algorithms for finding the shortest path in several routing problems. An ACO algorithm includes a stochastic local search strategy to structure the routing paths which can be established by a set of artificial ants. These ants work cooperatively using indirect communication of information for construction of the optimal shortest path. Inan intelligent optimization algorithm, the ACO idea is borrowed from the food searching characteristic of the real ant colony and how ants do this difficult job when they work together. Depending on biological studies on ants, it can be assumed that the ACO performs the finding of the shortest path from the nest to the food. Ant's pheromone distribution mechanism to share information with other ants in indirect coordination is called stigmergy. A number of researchers have suggested that the ACO optimization algorithm is very good for collaborating, exchange and transmission of information. The ACO algorithm is based on pheromone updates. This pheromone updating depends on the best solution achieved by the pheromone amount and the number of ants. The natural ants find the shortest path based on their own best knowledge solution and it depends on a strong pheromone trace. Finding the shortest path using ACO algorithm is inversely proportional to the pheromone quantity and length of the path [10]-[12].

The ACO depends on a probabilistic method for solving the computational problems and minimizes the paths through graphs [13]. ACO algorithm can be given in detail as:

Let us assume

K = Number of ants

(Tau) ι_{ij} = The pheromone concentration with the edge

η_{ij} = The heuristic information (based on experience)

α = pheromone removal parameter

β = pheromone deposition parameter

ρ = Represents the evaporation rate to avoid accumulation of the pheromones

$\Delta\tau_{i,j}^{k}$ = Amount of pheromones needed to path travel by ant K

L_k = Total length travel by ants K

Q = Constant

$$p_{ij}^{k}(t) = \begin{cases} \dfrac{[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k \in allowedk}[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} & \text{If j } \varepsilon \text{ allowed k} \\ 0 & \text{Otherwise} \end{cases}$$

$\iota_{ij}(t+n) = \rho\iota_{ij}(t) + \Delta\iota_{ij}$

$$\Delta\iota_{i,j}^{k} = \begin{cases} \dfrac{Q}{L_k} & \text{If (i, j) } \varepsilon \text{ tour described by taboo k} \\ 0 & \text{Otherwise} \end{cases}$$

Clustering is used in a wide range of research areas like engineering, medicine, data mining, biology, artificial intelligence and even IoT. Xu and Wunsch (2005) have represented an abbreviated survey on clustering algorithms. K-means clustering is the most commonly used algorithm. K-means clustering algorithm divides the data/substances into a number of clusters based on minimizing the sum of the squared distances between the data/substances and the centroid of the clusters. The k-means clustering algorithm is one of the simplest algorithms, but it is not suitable for a large amount of data set due to higher time complexity. Various methods have been proposed to accelerate the working of k-means such as when computation complexity is increased the backtracking is required [4], [12].

In recent years, many new clustering algorithms have been proposed

after deep study on clustering. There is a clustering algorithm which is based on ant system. A combination of two or three different clustering algorithms is new to clustering analysis. Clustering analysis plays an important role in the datamining field. Data can be grouped into different classes or clusters by clustering analysis. There exists better similarity among the objects in the same class and poorer similarity among the objects in different classes. In machine learning, clustering is a kind of unsupervised learning because it has no prior knowledge of classification labels. Clustering analysis is widely applied in image processing, model recognition, document retrieval, medical diagnosis, web analysis etc. [4], [14].

A. The Basic Principle of the K-means Algorithm

K objects are randomly chosen from n objects as initial clustering centers. Then the algorithm calculates the distance from each object to k clustering centers and judges which clustering center is nearest, assigning the object to the cluster of the nearest center. When all the computation work is done, it will form knew clusters. Next, the algorithm re-computes a mean value of each new cluster as its new clustering center. According to the above procedure, the algorithm will repeat calculating the distance and iterating till criterion function converges. The sum of square error is often-used as the criterion function. It is defined as [4], [14]:

$$E = \sum_{i=1}^{k} \sum_{p=Ci} |p - mi|2 \tag{1}$$

E is the sum of square error to all objects in the database. p is a point in the space that expresses a given object. m_i is the mean of clustering C_i. According to this criterion, data belonging to the same class are as similar as possible and data from different classes are as different as possible [4], [14].

The article is divided into six sections. After a brief introduction about ACO and k-means clustering in section I, section II contains the related work. In section III we explain our problem definition, in section IV we show our proposed algorithm, and finally, section VI explains the discussion and conclusions.

II. Related Work

The IoT environment contains a large scale of different types of networks. Routing techniques in WSN from source to destination is one of the important issues in the IoT system. The algorithms which are used to select the cluster heads/nodes depend on specific characteristics of clusters and/or network environments like energy level that suffers from complexity. Hence, the architectures of IoT are unsuitable and features of IoT application are dynamic. A large number of different kinds of research work on routing in IoT from source to destination have been done in literature [1], [4], [26].

Omar Sajid [1] has proposed optimum routing path using Ant Colony Optimization (ACO) algorithms inside the IoT system. Depending on the types of network, Sajid [1] has suggested to divide the IoT environment into various zones like status, requirements, etc. then use the ACO algorithm that was fit for each network. Finally, the simulation results proved that the proposed routing algorithm has better energy saving techniques. Kumar et al. [4] has presented a comparison of some clustering algorithms to analyze the scheduler performance and suggested that K-means based clustering is effective for the IoT based environment. Lu et al.[10] has suggested that the ACO finds the path to broadcast signaling contained in various network nodes and various flexible network structure problems, and during simulation analysis it is noticed that finding the path by ACO in IoT decreases the transmission storm efficiently. When the number of nodes increases in the finding path process, then it is important to reduce the time of path structure. In order to analyze a large-scale routing strategy, Guang Ji [15] has

proposed IoT ant colony searching routing based on Markov decision model. Markov decision ant colony routing selection algorithm is based on multi-parameter equilibrium. Markov routing is a decision model to estimate the number of nodes in a node communication range and facilitates the decision which meets the requirements. This algorithm efficiently decreases the overhead workload which is generated by controlled messages and multiple hops routing between clusters and make the evaluation function value of the path for allowing decision set corresponding to the pheromone concentration of ant colony for repeating the process. During the routing discovery phase, it calculates the transition probability of nodes and selects the global optimal routing. During the simulation-based analysis, it is observed that the problem of network "hot spots" is effectively solved by the Markov-A algorithm and the energy consumption of the network is balanced so that the life cycle of the network is prolonged. Dorigo et. al. [16] has specified an explanation of the Ant Colony Optimization (ACO) meta-heuristic and has discussed the type of problems where it can be applied. Dorigo et. al. [16] has used ACO algorithms in two typical applications, namely traveling salesman problem and routing in packet-switched networks. Merkle et. al. [17] has introduced an ACO method for the resource-constrained project scheduling problem (RCPSP). It is a combination of direct (or local) and indirect (or global) for ants in the structure of a new solution and uses pheromone evaluation approaches. From newly added features, this algorithm changes the strength of the heuristic effect and the rate of pheromone vaporization over ant peers. Below some limitation author's proposed algorithm perform the best solution compared to some other heuristics with and without limitations to the number of evaluated schedules shows the flexibility of the method. Michael Frey et al. [18] have proposed a framework and methodology to study ant routing algorithms for wireless networks. While running experiments in a wireless test bed is a number of some, expensive and error-free task, studying ant routing algorithms in simulation allows investigating some specifics properties of these algorithms more easily. This includes behavior of all aspects such as adaptive and pheromone evolution, the scalability in respect to the number of nodes or traffic flows, and mobile scenarios. These frameworks are easy to extend and customize by providing new back ends for different network simulators (or test bed frameworks) which is feasible with acceptable efforts. Mariusz et al. [19] have proposed Ant Colony Optimization (ACO) based algorithm designed to find the shortest path in a graph. The algorithm consists of several sub problems that are presented successively. Each sub problem is discussed from many points of view to enable researchers to find the most suitable solutions for the problems. Algorithms based on the metaheuristic of ant colony do not guarantee finding an optimal solution in all possible cases. Accordingly, to experimentation, it is particularly important to find out and select parameters dedicated to each of the problems under consideration. Individual elements of the procedures applied in the process should be also analyzed with regarding to their usability and purpose fullness of application. The construction of this Shortest Path ACO algorithm directly reflects to the various variants of the execution of individual elements of the procedure. In this way, it is possible to improve the method for the solution of the shortest path problem to approach or reach optimal solutions. An evaluation of the duration time and the quality of returned solutions will provide information for making a decision on the implementation of a given scheme as being of optimum quality or an alternative to more time-consuming procedures or procedures with higher computational cost. Yuq in get. al. [20] has recommended a new K-Means algorithm. This algorithm is a combination of density-based and ant searching theory, which is controlled by the initial parameter of k-means and local minimal by the random ants. The experiments analysis shows that, this algorithm has better quality for productivity and accuracy of clusters. Thus, it can place the similar types of objects together in one cluster and eliminate the dissimilar types of objects

away. This procedure has random competence of ACO which avoid clustering success into local optimality, and it furthermore avoids responsiveness of the primary partition of the k-means algorithm. Gelenbe et al. [25] has proposed the relationship and effect between choice of system load, energy consumption and QoS using a simple queuing model. They [25] have analyzed the parameters which are effect of response time of the system and energy cost per job using the mathematical queuing model. S. Kumar et Al. [30], [31] and V. García-Díaz et Al. [32] have proposed the Supply Chain Management based model for optimizing the response time and job scheduling by applying M/M/1 queuing model in IoT environment.

III. The Problem Definition

This section contains the problem statement followed by a description of the ACO and K-Means clustering based IoT messaging service architecture considered in the work and the job model.

A. Problem Statement

In an IoT environment, a big amount of heterogeneous wired and wireless devices/objects interconnect with each other identified by IPv6 addressing using single or multiple levels of subnets. These devices/objects generate a big amount of data (much time a continuous stream too) and scheduling these in the IoT environment from source to destination becomes a challenging issue.

IoT is a mixture of multiple wired or wireless communication technologies. The routing is the most important challenge in the IoT environment for solving how to find the best optimal path for data transmission from one node to another node in a different environment. An IoT environment includes different types of networks which depend on the network's status, and requirements. In this article, it is proposed that each network has own responsibility for finding an optimal path in the IoT environment. There are many inter-connections between different networks and the seinter-connections are called overlapped areas. So, this work intends creating an algorithm to control the use of algorithms and determines a solution for overlapped areas problem. This algorithm has been tested and compared with ACO and K-Means clustering algorithms that are closely related to the IoT routing problem.

The prime objective of this work is to calculate and compare the response time for message forwarding of the entire IoT environment using ACO and k-means clustering algorithms approaches to find the suitable path for reducing the energy consumption/cost. In the big transportation for IoT background, this is very beneficial for providing the flexible and effective response time services. The number of clusters is one of the most important features for calculating the performance of K-means clustering algorithm and the number of paths is one of the most important features for calculating the performance of ACO algorithm. The nodes (for ACO algorithm) and Cluster Head (*CH*) (for K-means clustering algorithm) may be completely connected or partially connected. The processing speed of each *CH/Node* can be measured as MIPS (Million Instruction Per Cycle) count [4], [21], [26].

B. Job Model

Each job or message is divided into sub-jobs/tasks depending on the priority of jobs and sequence of data messages. Data are available in the format of data packets and need the transmission from source to destination. The jobs are mathematically modeled by a weighted graph $D= (T, E)$ where T shows the set of t tasks and E shows as the set of e edges among the jobs. The edges show the priority of task/message [4], [26].

C. A Route Planning Model for K-Means Clustering and ACO Algorithms

The performance measurement for both K-Means clustering algorithms and ACO algorithms is done in terms of the average response time. It is assumed that each server follows the *M/M/1* queuing model where λ is the arrival rate and μ is the service rate [4], [22], [26].

The set of objects in different colors represents them belonging to different clusters and paths. A sample shortest path is shown between source to destination base station based on a hypothetical approach.

QoS parameters like Average Response Time (*RT*), Average Waiting Time (*WT*), and Average Queue Length (*QL*) have been estimated. Here, it is assumed that each server follows the *M/M/1* queuing model. The average queue length at the i_{th} *CH/Node* with both the number of jobs waiting in the queue and those in service can be written as $E[Ni] = \frac{\rho_i}{(1-\rho_i)}$ Where ρ_i is the i_{th} server utilization and whose average queue length is $E[Ni]$ [4], [23]-[24], [26].

The average response time depends on how to quick response a *CH/Node*. Arrival rate of the jobs at the input queue at each *CH/Node* is randomly generated. The average response time can be estimated as [4], [23]-[24], [26]:

$$\text{Average Response Time} = \frac{AveregeQueueLength}{ArrivalRate} \qquad (2)$$

The waiting time is the period of time where the job does not execute because of the execution preference or some event to happen. So, the average waiting time at each CH/Node in the path can be calculated as [4], [23]-[24], [26]:

$$\text{Average Waiting Time} = AverageResponseTime - \frac{1}{ServiceRate} \qquad (3)$$

Fig. 1, represents a short example of the proposed IoT routing algorithm model using ACO algorithm.
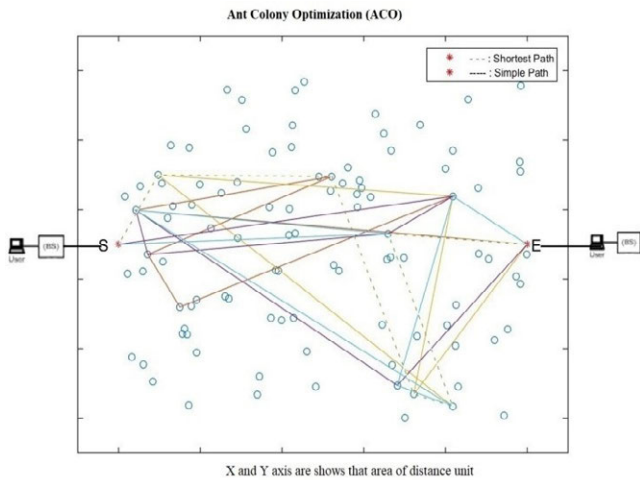

Fig. 1. Architecture of IoT Network Using ACO Algorithm and Messaging Forwarding.

Let us assume that there are *n* nodes in a network and *m* nodes transmitting the signal for searching at the same time in network routing. $\iota_{ij}(t)$ is the number of active signaling established through the path among node *i* and *j*, d_{ij} (*i, j = 1, 2,...,n)*stands for the distance among the node *i* and *j* at the time *t*.

In the initiation stage, m random nodes are selected, the number of active signaling among the nodes *i* and *j* are $\iota_{ij}(0)$ and *ι (tabu)* is the primary part of individual signaling. *k* is allocated as a preliminary node.

Here, ι_{ij}^{k} (*t*) stands for the probability of *k* transmitting signal from node *i* to node *j* at the time of *t*, then:

$$p_{ij}^{k}(t) = \begin{cases} \frac{[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}}{\sum_{k \in \text{allowed } k}[\iota_{ij}(t)]^{\alpha}[\eta_{ij}(t)]^{\beta}} & \text{If } j \in \text{ allowed } k \\ 0 & \text{Otherwise} \end{cases} \qquad (4)$$

The permitted *k= {0, 1... ,n - 1}*, and ι^{st} and for the set of nodes and indicating next permitted node to pass. The difference between an artificial ant and real ant colony is the capability of memory. The ι (*k = 1, 2,...,m*) is utilizing the records of nodes and signaling *k* pass over at the current time, and it is dynamically adjusted with the transmission of signaling *k* process. *1-ρ*representstheunit of disappearing and*α, β*individually represent the amount of information collection of signaling in the process of re-transmission. It plays different roles for the heuristic aspect in the path selection through the signaling re-transmission. $\eta_{ij}(t)$ represents the predictions unit of the transmission between node *i* to *j*. Then signaling *k* cover all nodes and make a complete cycle. At that time, the information of all paths is updated according to the following equation:

$$\iota_{ij}(t+n) = \rho\, \iota_{ij}(t) + \Delta\, \iota_{ij} \qquad (5)$$

$$\text{Then } \Delta\, \iota_{ij} = \sum_{k=1}^{m} \Delta\iota_{ij}^{k} \qquad (6)$$

$\Delta\iota_{ij}^{k}$ represents the amount of information of signaling *k's* suggestions between node *i* and *j* in this cycle.

$$\Delta\iota_{i,j}^{k} = \begin{cases} \frac{Q}{L_k} & \text{if signaling k passes through the node i and j in the cycle} \\ 0 & \text{Otherwise} \end{cases} \qquad (7)$$

Here, *Q* represents the constant, L_k represents the path length that signaling k has paced in this cycle.

In order to estimate the shortest path using ACO from a source node (*S*) to a destination (*E*) it is assume that $\lambda_1$isthe arrival rate of the job at *Node_1*, working as a source node (*S*) for transmission to the destination (*E*)*Node_5*. *Node_1* has nodes *Node_2*, *Node_3*, and Node_4 as immediate neighbors. Let the probability of arrival of jobs at queues of these neighbors be P_2, P_3, and P_4 respectively such that $P_2+P_3+P_4=1$. In this article it is assumed that there are equal probabilities for selection of all the paths i.e. $P_2=P_3=P_4$. After being serviced by server *Node_1*, the jobs arrive at *Node_2* being the best path among the available paths offering the minimum response time with probability P_3. Let this probability be referred to as $P_1$for the remaining path, indicating the path chosen in the beginning. Therefore, the arrival rate for *Node_3* can be written as $\lambda_1 P_3 = \lambda_1 P_1$. Similarly, after being serviced by server *Node_3*, the jobs arrive at *Node_4* followed by *Node_3* to finally reach *Node_5* with arrival rate $\lambda_1 P_1$.

Let the service rate of *Node_1*, *Node_2*, *Node_3*, *Node_4*, and *Node_5*beμ_1, μ_2, μ_3, $\mu_4$and μ_5 respectively.

Therefore, the utilization of *Node_1* with $P_1=1$ becomes

$$\rho_1\,(Node_1) = \frac{\lambda_1}{\mu_1} \qquad (8)$$

The utilization of the selected nodes can be written as [4], [23]-[24], [26]

$$\rho_j\,(Node_j) = \frac{\lambda_1 P}{\mu_j} \qquad (9)$$

Where *i* = Number of stages in the network and

j = Index of the selected node in each stage.

The average queue length for *Node_1* becomes

$$E\,[QL_1{}^{(Node_1)}] = \frac{\rho_1(Node1)}{(1-\rho_1\,(Node1))}(\text{Using } \rho_1\,(Node_1) \text{ value from Eq. (8)})$$

$$= \frac{\frac{\lambda_1}{\mu_1}}{1-\frac{\lambda_1}{\mu_1}} = \frac{\frac{\lambda_1}{\mu_1}}{\frac{\mu_1-\lambda_1}{\mu}} = \frac{\lambda_1}{\mu_1-\lambda_1} \qquad (10)$$

Similarly, the average queue length can be calculated for the other selected nodes in the path as [4], [23]-[24], [26]

$$E\ [QL_j{}^{(Node_j)}] = \frac{\rho_{1j}(Node_j)}{(1-\rho_j\ (Node_j))} = \frac{\lambda_1 P}{\mu_j-\lambda_j} \qquad (11)$$

The queue length for the entire path can be estimated as:

$E\ [QL_{avg}]$ = Sum of Average Queue Length (QL) value of selected nodes in every stage.

Therefore,

$$E\ [QL_{avg}] = \sum E\lfloor QL_i^{Node_j}\rfloor \qquad (12)$$

Here 'i' is the stage and $Node_j$ is the node being selected for message forwarding.

$$Average\ Response\ Time = \frac{AveregeQueueLength}{ArrivalRate}(\text{from equation (2)})$$

In the current case, the path from $Node_1$ (source) to $Node_5$ (destination) comprises of $Node_2$, $Node_3$, and $Node_4$. The average response time of for $Node_1$:

$$E\ [RT_1{}^{(Node_1)}] = \frac{E\ [N_1(Node1)\ ]}{\lambda_1}(\text{using } E\ [QL_1\ (Node_1)]\text{value from Eq.}(10))$$

$$= \frac{\frac{\lambda_1}{\mu_1-\lambda_1}}{\lambda_1} = \frac{1}{\mu_1-\lambda_1} \qquad (13)$$

For $Node_j$ server the response time can be written as:

$$E[RT_i{}^{(Node_j)}] = \frac{\rho_i(Node_j)}{(1-\rho_i\ (Node_j))} = \frac{1}{\mu_j-\lambda_1 P} \qquad (14)$$

The average response time for the paths selected become the sum of the Average Response Time (RT) values of selected $Node_j$ in every stage 'i'.

$$E[RT_{avg}] = \sum E\lfloor RT_i^{Node_j}\rfloor \qquad (15)$$

$$Average\ Waiting\ \ Time = AverageResponseTime - \frac{1}{ServiceRate}$$
(from Eq. (3))

$E\ [WT_1{}^{(Node_1)}] = E\ [R_1{}^{(Node_1)}]$ - $\frac{1}{\mu_1}$ (using $E\ [RT_1\ (Node_1)]$ value from Eq.(13))

$$= \frac{1}{\mu_1-\lambda_1} - \frac{1}{\mu_1} = \frac{\mu_1-\mu_1+\lambda_1}{(\mu_1-\lambda_1)\mu_1} = \frac{\lambda_1}{(\mu_1-\lambda_1)\mu_1} \qquad (16)$$

Similarly, for $Node_j$ server the waiting time can be written as:

$$E\ [WT_i{}^{(Node_j)}] = \frac{\rho_i(Node_j)}{(1-\rho_i\ (Node_j))} = \frac{\lambda_1 P}{(\mu_j-\lambda_1 P)\mu_j} \qquad (17)$$

The average waiting time for the complete path can be written as the sum of the average Waiting Time (WT) values of selected nodes in every stage as:

$$E[WT_{avg}] = \sum E\lfloor WT_i^{Node_j}\rfloor \qquad (18)$$

Fig. 2 represents a short example of a proposed IoT routing algorithm model using K-Means clustering algorithms.

In order to use the K-Means clustering algorithms, a sample path chosen among the clusters to route the messages from Source (S) to Destination (E) has been presented and routing paths as per the K-Means clustering algorithmic characteristic and some properties like Euclidean distance and Degree of nodes to find the shortest path from Source (S) to Destination (E).

Here it is assumed that P is the probability of arrival of the jobs at the source queue to be forwarded to the destination in K-Means clustering through Cluster Head (CH). Let the arrival rate of the jobs at CH_1 be λ_1 which is the Start CH 'S' in the IoT network. The packet needs to be transmitted to the end cluster 'E' which is CH_5 which can be the preferred endpoint like cloud storage. After being serviced by CH_1, let the message be forwarded to CH_2. As discussed above, the arrival rate

for CH_2 becomes $\lambda_2 P = \lambda_1 P$. Similarly, the message reaches destination CH_5 with arrival rate $\lambda_1 P$, being routed through various cluster heads CH forming the path.
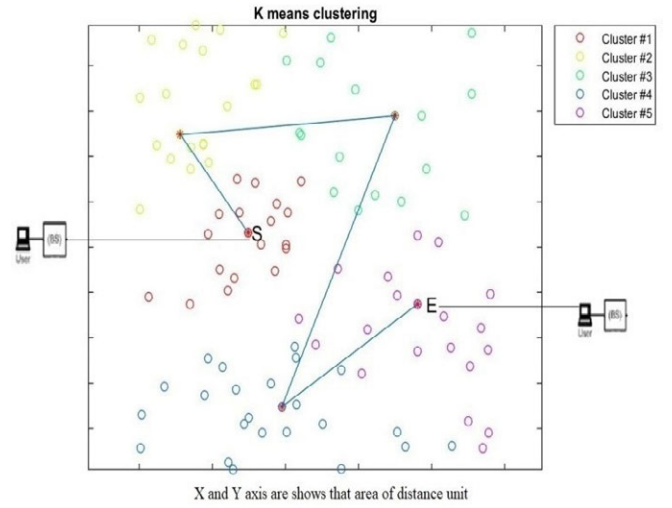

Fig. 2. Architecture of IoT Network Using K-means Clustering Algorithm and Messaging Forwarding.

The best and stable utilization of CH_1 server is [4], [22], [26]

$$\rho_1(CH_1) = \lambda_1/\mu_1 \qquad (19)$$

Similarly, for CHj server, the utilization can be written as

$$\rho_j(CH_j) = \lambda_1/\mu_j \qquad (20)$$

where i = Number of stages in the network and

j = Index of selected node in each stage.

Therefore, the utilization of $Node_1$ with $P_1 = 1$ becomes $\rho_1(Node_1) = \frac{\lambda_1}{\mu_1}$ from Eq.(8).

The average queue length for CH_1 [4], [22], [26]:

$$E\ [QL_1(CH_1)] = \frac{\rho_1(CH1)}{(1-\rho_1\ (CH1))}\ (\text{using } \rho_1(CH_1)\text{ value from Eq.}(16))$$

$$= \frac{\frac{\lambda_1}{\mu_1}}{1-\frac{\lambda_1}{\mu_1}} = \frac{\frac{\lambda_1}{\mu_1}}{\frac{\mu_1-\lambda_1}{\mu_1}} = \frac{\lambda_1}{\mu_1-\lambda_1} \qquad (21)$$

Similarly, the average queue length can be calculated for the other selected CH_j in the path as [4], [23]-[24], [26]:

$$E\ [QL_j{}^{(CH_j)}] = \frac{\rho_j(CHj)}{(1-\rho_j\ (CHj))} = \frac{\lambda_1 P}{\mu_j-\lambda_j} \qquad (22)$$

The queue length for the entire path can be estimated as:

$E\ [QL_{avg}]$ = Sum of Average Queue Length (QL) value of selected CH in every stage.

Therefore,

$$E\ [QL_{avg}] = \sum E\lfloor QL_i^{CHj}\rfloor \qquad (23)$$

Where 'i' is the cluster and CH_j is the node being selected for message forwarding.

$$Average\ Response\ Time = \frac{AveregeQueueLength}{ArrivalRate}\ (\text{from Eq. (2)})$$

$$E\ [RT_i(CH_1)] = \frac{E\ [QL1(CH1)]}{\lambda_1}(\text{using } E[QL_1(CH_1)]\text{value from Eq. (21)})$$

$$= \frac{\frac{\lambda_1}{\mu_1-\lambda_1}}{\lambda_1} = \frac{1}{\mu_1-\lambda_1} \qquad (24)$$

Similarly, average response time can be calculated for the other selected CH_j in the path as [4], [23]-[24]:

$$E[RT_i^{(CHj)}] = \frac{\rho_i(CHj)}{(1-\rho_i(CHj))} = \frac{1}{\mu_j - \lambda_1 P} \quad (25)$$

The average response time for the paths selected become the sum of the Average Response Time *(RT)* values of selected CH_j in every stage '*i*'.

$$E[RT_{avg}] = \sum E\lfloor RT_i^{CHj} \rfloor \quad (26)$$

$Average\ Waiting\ Time = AverageResponseTime - \frac{1}{ServiceRate}$
(from Eq. (*3*))

$E[WT_l^{(CH_l)}] = E[R_l^{(CH_l)}] - \frac{1}{\mu_1}$ *(using* $E[RT_l(CH_l)]$ value from Eq.*(24))*

$$= \frac{1}{\mu_1 - \lambda_1} - \frac{1}{\mu_1} = \frac{\mu_1 - \mu_1 + \lambda_1}{(\mu_1 - \lambda_1)\mu_1} \quad (27)$$

Similarly, for CH_j server the waiting time can be written as:

$$E[WT_i^{(CHj)}] = \frac{\rho_i(CHj)}{(1-\rho_i(CHj))} = \frac{\lambda_1 P}{(\mu_j - \lambda_1 P)\mu_j} \quad (28)$$

The average waiting time for the complete path can be written as the sum of the average Waiting Time (*WT*) values of selected *CH* in every stage as:

$$E[WT_{avg}] = \sum E\lfloor WT_i^{CHj} \rfloor \quad (29)$$

IV. The Algorithm

The primary objective of this algorithm is to calculate and compare the message response time from source to destination by preserving the service quality of fluctuating sized messages from the source to the destination through *CHs/Nodes* in the network. The word Cluster Head (*CH*) for K-Means clustering algorithms and the word Node for ACO Algorithms are used. When a sensor device sends a message from the source *CH/Node* to the destination *CH/Nodes*, it fritters away some communication cost in the transfer of the message. In this procedure, it waits for the response of each *CH/Node* in each IoT network environment. This is a big problem for IoT because it may affect the battery life of the sensor *CH/Node* as the battery life gets weak with every waiting period. However, if this response time was reduced by proper path choice, it would decrease the waiting time and eventually preserve the power of the sensor's devices. Thus, this method decreases the response time and makes it an energy-aware scheduling algorithm too.

These messages convey data with respect to the physical parameters which change continuously in some degree. Algorithm 1 corresponds to the message-scheduling algorithm for the IoT framework.

V. Performance Evaluation

To evaluate the performance of QoS parameters, Matlab64-bit version *8. 5.0.197613 (R2013a)*, processor Intel *(R) Core (TM) i7-4790 CPU @ 360GHz, 64*-bit Operating System, and RAM *4 GB* as a simulation platform has been used. *M/M/1* priority queuing model has been used for resource provisioning and message scheduling system within the IoT environment. Applied *M/M/1* queuing model can be reconsidered for some newer models like phase-type queuing networks or Pareto-distribution-based queuing networks. The basis of considering the *M/M/1* queuing model is that it considers each *node/CH* as a single node and has been established to fit well in these types of problems. Random data values have been generated for the experiment during execution.

Algorithm 1: Response Time Aware Scheduler

Initialize:

  Submit the message request

  Identify the Source 'S' and the Destination 'E'

  Record the Arrival rate of the batch of jobs/message

  For every Source Node/CH,

    Evaluate the number of messages/ modules

    Sort the job/messages according to priority

Establish the Queues:

  For the Source 'S', do,

    Assign job/message to the high priority/low priority queue

Route Selection:

  For the source node, select the nearest node using ACO and K-means approach as per Section 3

  For the specified job/message

  Evaluate the execution time of every job/message at individual CH/Node.

  If a CH/Node is Busy then // Upcoming message is waiting

  For i = 2: N // Following messages

    Message Wait (i) = Job Execution Time – Priority Message (i, 1);

  End for

    Calculate Average Queue Length // using eq. 12 (For ACO) and 23 (For K-means)

    Calculate Average Response Time// using eq. 15 (For ACO) and 26 (For K-means)

  Calculate Average Waiting Time// using eq. 17 (For ACO) and 29 (For K-means)

Fig. 3 represents the Response time of ACO and K-means clustering algorithm for a fixed job size of *1000* and varying the CHs (For K-means) and Node (For ACO) from *10* to *50* with the numeric data represented in Table I. Further, ACO Response time result is a minimum and better than the K-means Clustering Algorithm response time.
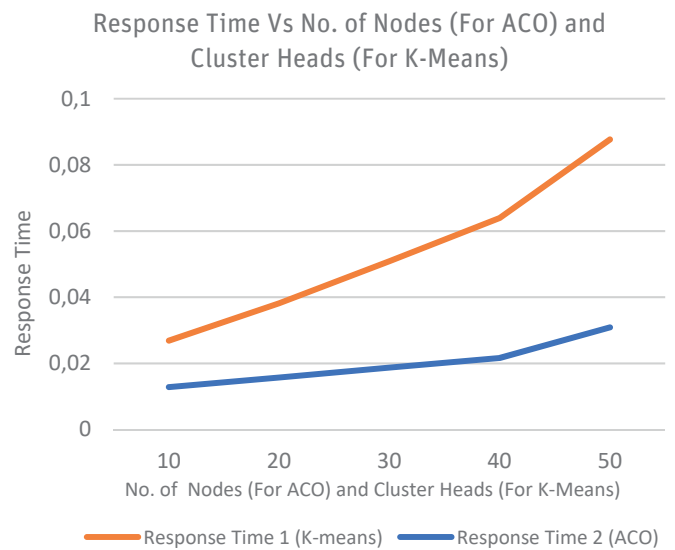

Fig. 3. Batch of Jobs Response Time fora fix jobs size and varying the Nodes and Cluster Heads (CHs).

TABLE I. Response Time for a Fix Jobs Size and Varying the Nodes and Cluster Heads (CHs) Corresponding to Fig. 3

No. of Cluster Heads	Response Time (ACO Algorithm)	Response Time (K-means Clustering Algorithm)
10	0.0128524	0.0140297
20	0.0157312	0.0224375
30	0.018759	0.0321506
40	0.02162207	0.042322676
50	0.03088497	0.056817971

The effect of some other QoS parameters such as the Average Service Rate, Average Queue Length and Average Waiting Time are shown in Fig. 4-6.In this experimental setup, we fixed the job size of *1000* while varying the number of CHs (For K-means) and Node (For ACO)from *10* to *50*. The numeric data is represented in Table II-IV. Further, ACO results are better than the K-means Clustering Algorithm.
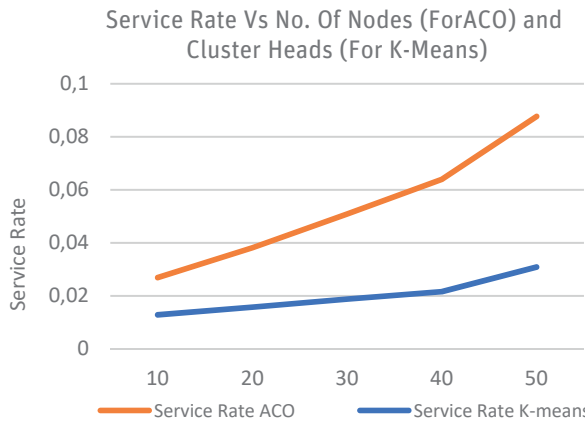

Fig. 4. Batch of Job's Service Rate for a fixed job size and varying Nodes and Cluster Heads (CHs).

TABLE II. Service Rate for A Fixed Job Size and Varying Nodes and Cluster Heads (CHs) Corresponding to Fig. 4

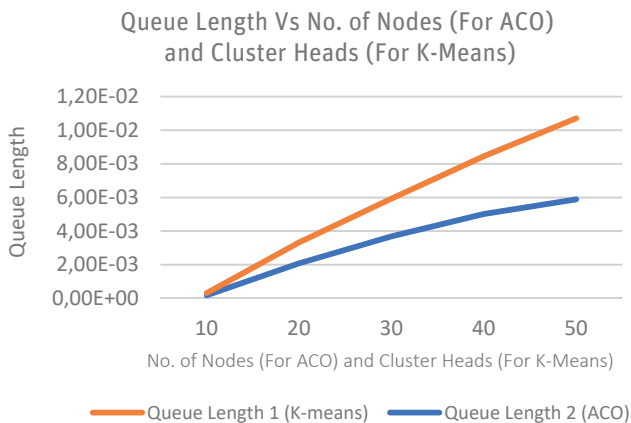| No. of Cluster Heads	Service Rate (K-means Clustering Algorithm)	Service Rate (ACO Algorithm)
10	371.1397	644.75827
20	734.84904	963.0289
30	946.64061	1501.5987
40	1254.2681	2209.5784
50	1696.58578	3338.42747

Fig. 5. Batch of Job's Queue Length for a fixed job size and varying Nodes and Cluster Heads (CHs).

TABLE III. Queue Length for a Fixed Job Size and Varying Nodes and Cluster Heads (CHs) Corresponding to Fig. 5

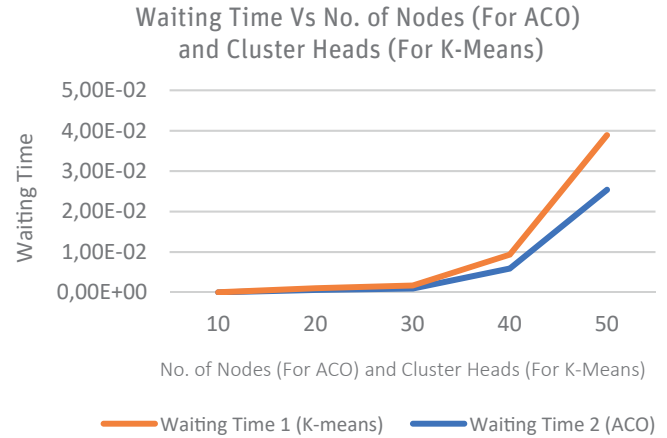| No. of Cluster Heads	Queue Length (ACO Algorithm)	Queue Length (K-means Clustering Algorithm)
10	0.000131	0.00018593
20	0.00125	0.00208
30	0.00225	0.00368118
40	0.003433	0.00502
50	0.00482	0.00589

Fig. 6. Batch of Job's Waiting Time for a fixed job size and varying Nodes and Cluster Heads (CHs).

TABLE IV. Waiting Time for a fix Job Size and Varying the Nodes and Cluster Heads (CHs) Corresponding to Fig. 6

No. of Cluster Heads	Waiting Time (ACO Algorithm)	Waiting Time (K-means Clustering Algorithm)
10	0.0000122	0.0000145
20	0.000414	0.000614
30	0.000745	0.000962
40	0.00342	0.00592
50	0.0135	0.0254

Fig.7 represents the effect of both ACO and K-means clustering algorithm Response time when we vary the job size from *200* to *1000* for a fixed number of Nodes (For ACO) and CHs (For K-means) that is *10*, with the numeric data represented in Table V. We observed that the response time of ACO is again better than the K-means Clustering Algorithm.
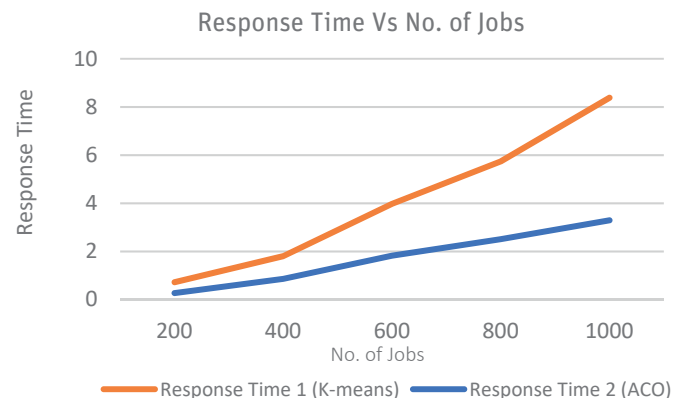

Fig. 7. Batch of Job's Response Time for varying No. of jobs and fixed Nodes and Cluster Heads (CHs).

TABLE V. Response Time for No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig. 7

No. of Jobs	Response Time (ACO)	Response Time (K-means Clustering Algorithm)
200	0.2714044	0.4507078
400	0.86546546	0.938945
600	1.826384	2.1509484
800	2.5126439	3.2269072
1000	3.293245	5.0905315

Fig. 8. Batch of Job's Service Rate for varying No. of jobs and fixed Nodes and Cluster Heads (CHs).

TABLE VI. Service Rate for varying No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig.8

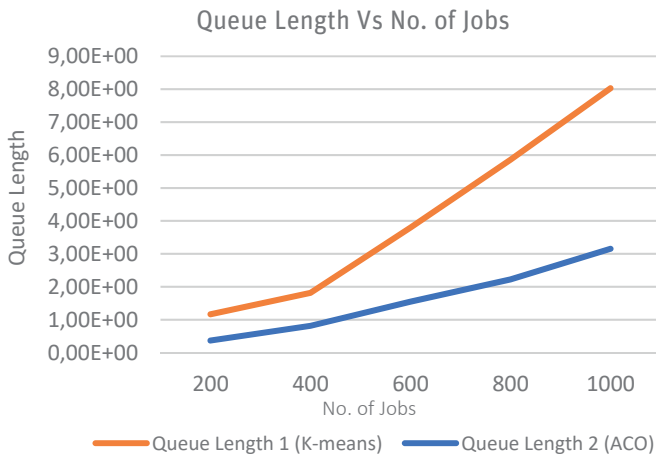| No. of Jobs	Service Rate (K-means Clustering Algorithm)	Service Rate (ACO Algorithm)
200	224.4072	616.66515
400	479.8531	913.67953
600	714.88793	1480.2342
800	1046.12343	2279.5486
1000	1552.5966	3425.1479

Fig. 9. Batch of Job's Queue Length for varying No. of Jobs and fixed Nodes and Cluster Heads (CHs).

TABLE VII. Queue Length for Varying No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig. 9

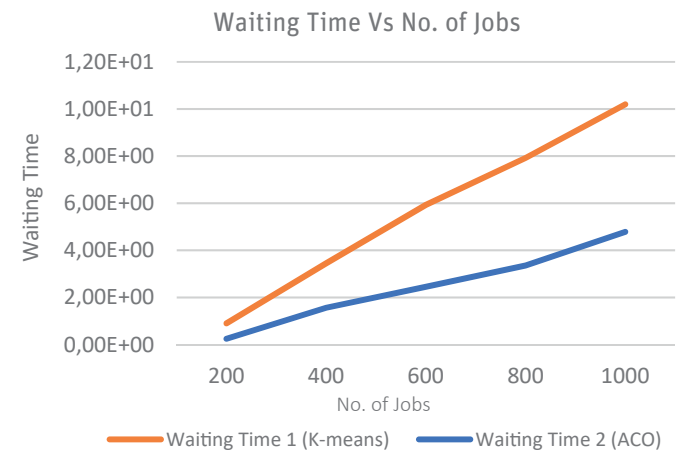| No. of Jobs	Queue Length (ACO Algorithm)	Queue Length (K-means Clustering Algorithm)
200	0.36539	0.800113
400	0.815	1.00149
600	1.550283	2.24586
800	2.223213	3.63284
1000	3.15293	4.876906

Fig.10. Batch of Job's Waiting Time for varying No. of jobs and fixed Nodes and Cluster Heads (CHs).

TABLE VIII. Waiting Time for Varying No. of Jobs and Fixed Nodes and Cluster Heads (CHs) Corresponding to Fig. 10

No. of Jobs	Waiting Time (ACO)	Waiting Time (K-means Clustering Algorithm)
200	0.2474654	0.654986
400	1.56216	1.89234
600	2.45559107	3.478343
800	3.354562	4.564951
1000	4.784304	5.4124101

VI. Conclusion

IoT is poised to change the way of living. With huge heterogeneity and dynamicity in IoT, the response time should be ensured to be as low as possible for better network performance leading to an efficient and smart IoT. The response time has effect on the use of energy cost per job for the system [25]. The optimization of the response time for transmission of data/jobs in the entire system of IoT environment automatically optimizes the cost/use of energy. This work contains a comparative analysis between ACO and K-means algorithms based on a job/message scheduling model for IoT, based on an *N* layered network. Performance measures such as the average queue length, average waiting time and average response time have been derived, plotted and analyzed. It is noticed that the ACO offers better performance for the considered parameters. The model gains significance due to the fact that efficient message forwarding in IoT will ensure the optimum use of sensor energy to realize a truly smart framework.

References

[1]   O. Said, "Analysis, design and simulation of Internet of Things routing algorithm based on ant colony optimization", International Journal of Communication Systems, Wiley, 2016.

[2]   S. Kumar, Z. Raza, "Internet of Things: Possibilities and Challenges", International Journal of Systems and Service-Oriented Engineering (IJSSOE), Vol.7, no.3, pp. 32-52, July-September 2017 (ISSN 1947-3052).

[3]   K. Kumar, S. Kumar, O. Kaiwartya, Y. Cao, J. Lloret, N. Aslam, "Cross-Layer Energy Optimization for IoTEnvironments: Technical Advances and Opportunities", Energies, 2017.

[4]   S. Kumar, Z. Raza, "Using Clustering Approaches for Response Time Aware Job Scheduling Model for Internet of Things(IoT)", International Journal of Information Technology, Springer, Vol. 9, no. 2, pp. 177-195, June 2017.

[5]   S Vimal, M Khari, N Dey, RG Crespo, YH Robinson, "Enhanced resource allocation in mobile edge computing using reinforcement learning based MOACO algorithm for IIOT", Computer Communications 151, pp. 355-364, 2020

[6]   E. Dave, "The Internet of Things: How the Next Evolution of the Internet Is Changing Everything", Cisco Internet Business Solutions Group, pp. 1-11, April 2011.

[7]   Dr. Vermesanovidiu, Dr. F. Peter, G.Patrick, G.Sergio, S. Harald, Dr. B. Alessandro, J. Ignacio, Dr. M. Margaretha, Dr. H. Mark, Dr. E. Markus, Dr. D. Pat, "Internet of Things- Global Technological and Societal Trends", published by: River Publishers, ISBN: 9788792329677.

[8]   A. C. Charu, A.Naveen, S. Amit, "The Internet of Things: A Survey from the Data-Centric Perspective", Springer Science+Business Media New York, 2013, pp 383-428, ISBN 978-1-4614-6309-2.

[9]   A. Saima, Y. Kun "A QoS Aware Message Scheduling Algorithm in Internet of Things Environment", IEEE Online Conference on Green Communications (Online Green Comm), 2013.

[10]   Y. Lu, W. Hu, Study on the Application of Ant Colony Algorithm in the Route of Internet of Things", International Journal of Smart Home, Volume 7, No. 3, May, 2013.

[11]   S. Saatchi, C. C. Hung, "Hybridization of the Ant Colony Optimization with the K-Means Algorithm for Clustering", Springer, pp. 511 – 520, 2005.

[12]   C.I. Mary, DR. S.V. K. Raja, "Refinement of Clusters from k-means with Ant Colony Optimization", Journal of Theoretical and Applied Information Technology, 2009.

[13]   M. Khari, et al., "Performance analysis of six meta-heuristic algorithms over automated test suite generation for path coverage-based optimization", Soft Computing, In Press, 2019.

[14]   J. Lu, R. Hu, "A new hybrid clustering algorithm based on K-means and ant colony algorithm", Proceedings of the 2nd International Conference on Computer Science and Electronics Engineering (ICCSEE 2013).

[15]   C. Cheng, Z. Qian, "An IoT Ant Colony Foraging Routing Algorithm Based on Markov Decision Model", International Conference on Soft Computing in Information Communication Technology (SCICT 2014).

[16]   M. Dorigo, G. D. Caro, "Ant Colony Optimization: A New Meta-Heuristic", IEEE, 1999.

[17]   D. Merkle, M. Middendorf, H. Schmeck, "Ant Colony Optimization for Resource-Constrained Project Scheduling", IEEE, Transactions on Evolutionary Computation, Volume 6, No. 4, August, 2002.

[18]   M. Frey, M. Günes, "Attack of the Ants: Studying Ant Routing Algorithms inSimulation and Wireless Testbeds", arXiv:1409.0988v1 [cs.NI] 3 Sep 2014.

[19]   M.Głabowski, B.Musznicki, P. Nowak, P. Zwierzykowski, "Shortest Path Problem Solving Based on Ant Colony Optimization Metaheuristic", Image Processing & Communication, Volume 17, No. 1-2, pp. 7-18, 2017.

[20]   P. Yuqing, H. Xiangdan, L. Shang, "The K-means Clustering Algorithm Based on Density and Ant Colony", IEEE Int. Conf. Neural Networks & Signal Processing, Nanjing, China, December 14-17, 2003.

[21]   B. Mamalis, D. Gavalas, C. Konstantopoulos, G. Pantziou, "Clustering in Wireless Sensor Networks," Zhang/RFID and Sensor Networks AU7777. Proof Page 323 2009-6-24, 2012.

[22]   R.Jain, "Introduction to Queueing. In: The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation and Modeling," ch. 30. John Wiley & Sons, Inc., New York, 1991.

[23]   Trivedi, K. S., "Probability and Statistics with Reliability, Queuing and Computer Science Applications", Prentice Hall, 1982.

[24]   R. L. Graham, E.L. Lawler, J.K. Lenstra, A.H.G.R. Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey, Annals of Discrete Mathematics", Volume 5, pp 287–326, 1979.

[25]   E. Gelenbe, R. Lent, "Optimising Server Energy Consumption and Response Time", Theoretical and Applied Informatics, Volume 24, Issue, 4, pp 257-270, November 2012, ISSN: 1896-5334.

[26]   S. Kumar, Z. Raza, "A Priority Based Message Response Time Aware Job Scheduling Model for the Internet of Things (IoT)", International Journal of Cyber Physical System (IJCPS), IGI Global, Volume 1, Issue 1, pp 1-14, January-June 2019, ISSN: 2577-4867, DOI: 10.4018/IJCPS.2019010101.

[27]   C. G. García, E. R. N. Valdez, V. G. Díaz, C. P. G.Bustelo, J. M. C. Lovelle, "A Review of Artificial Intelligence in the Internet of Things, International Journal of Interactive Multimedia and Artificial Intelligence, Volume 5, Issue 4, pp 9-20, 2019.

[28]   C. G. García, D. M. Llorián, J. M. C. Lovelle, "A Review about Smart Objects, Sensors and Actuators", International Journal of Interactive Multimedia & Artificial Intelligence, Volume 4, Issue 3, 2017.

[29]   J. Molano, J. Lovelle, C. Montenegro, J. Granados, R. Crespo, "Metamodel for Integration of Internet of Things, Social Networks, the Cloud and Industry 4.0", Journal of Ambient Intelligence and Humanized Computing, Volume 9, Issue 3, pp 709-723, 2018.

[30]   S. Kumar, Z. Raza, "Using Supply Chain Management Approach for Message Forwarding for Internet of Things (IoT)", International Conference on Technology, Engineering and Science (IConTES), Volume 4, pp 21-27, 2018.

[31]   S. Kumar, Z. Raza, "A K-Means Clustering Based Message Forwarding Model for Internet of Things (IoT)", International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, pp 604-609, 2018.

[32]   V. García-Díaz et al., "TALISMAN MDE framework: an architecture for intelligent model-driven engineering", International Work-Conference on Artificial Neural Networks (IWANN), Springer, pp. 299-306, 2009.

Sumit Kumar

Sumit Kumar is currently an Assistant Professor in the Department of Information Technology, Gopal Narayan Singh University, Sasaram, Bihar, India. He received M. Tech. and PhD degrees in computer science from the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. He received MCA degrees in Ranchi University, Ranchi, India. His research interest is Internet of Things (IoT).

Vijender Kumar Solanki

Vijender Kumar Solanki, Ph.D., is an Associate Professor in Department of Computer Science & Engineering, CMR Institute of Technology (Autonomous), Hyderabad, TS, India. He has more than 11 years of academic experience in network security, IoT, Big Data, Smart City and IT. Prior to his current role, he was associated with Apeejay Institute of Technology, Greater Noida, UP, KSRCE (Autonomous) Institution, Tamilnadu, India & Institute of Technology & Science, Ghaziabad, UP, India. He has attended an orientation program at UGC-Academic Staff College, University of Kerala, Thiruvananthapuram, Kerala &Refresher course at Indian Institute of Information Technology, Allahabad, UP, India. He has authored or co-authored more than 50 research articles that are published in journals, books and conference proceedings. He has edited or co-edited 10 books in the area of Information Technology. He teaches graduate & post graduate level courses in IT at ITS. He received Ph.D in Computer Science and Engineering from Anna University, Chennai, India in 2017 and ME, MCA from Maharishi Dayanand University, Rohtak, Haryana, India in 2007 and 2004, respectively and a bachelor's degree in Science from JLN Government College, Faridabad Haryana, India in 2001.He is Editor in International Journal of Machine Learning and Networked Collaborative Engineering (IJMLNCE) ISSN 2581-3242, Associate Editor in International Journal of Information Retrieval Research (IJIRR), IGI-GLOBAL, USA,ISSN: 2155-6377 | E-ISSN: 2155-6385 also serving editorial board members with many reputed journals. He has guest edited many volumes, with IGI-Global, USA, InderScience& Many more reputed publishers.

Saket Kumar Choudhary

Saket Kumar Choudhary is an assistant professor in FCA, MRIIRS, Faridabad, Harayana, India. He has obtained his master degrees in Mathematics from the University of Allahabad, Allahabad, India in 2005, Master of Computer Application (MCA) from UPTU, Lucknow, India in 2010, Master of Technology (M.Tech) from Jawaharlal Nehru University, New Delhi, India in 2014. He is Ph.D (Computer Science and Technology) School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. Currently, he is working as an assistant professor in MRIIRS, Faridabad. His research interest includes mathematical modeling and simulation, dynamical systems, computational neuroscience: modeling of single and coupled neurons, computer vision, digital image processing, machine learning and artificial intelligence.

Ali Selamat

Ali Selamat has received a B.Sc. (Hons.) in IT from Teesside University, U.K. and M.Sc. in Distributed Multimedia Interactive Systems from Lancaster University, U.K. in 1997 and 1998, respectively. He has received a Dr. Eng. degree from Osaka Prefecture University, Japan in 2003. Currently, he is the Dean of Research Alliance in Knowledge Economy (K-Economy RA) UTM. He is also a professor at the Software Engineering Department, Faculty of Computing UTM. Previously he was an IT Manager at School of Graduate Studies (SPS), UTM. He is also a head of Software Engineering Research Group (SERG), K-Economy Research Alliance, UTM. He is the editors of International Journal of Digital Content Technology and its Applications (JDCTA), International Journal of Advancements in Computing Technology (IJACT) and International Journal of Intelligent, Information and Database Systems (IJIIDS). His research interests include software engineering, software agents, web engineering, information retrievals, pattern recognitions, genetic algorithms, neural networks and soft-computing.

Rubén González Crespo

Dr. Rubén González Crespo has received a PhD in Computer Science Engineering. He is also Industrial Engineering and MSc in Computer Science and Project Management. Currently he is Vice Rector of Academic Affairs from UNIR. He is advisory board member for the Ministry of Education at Colombia and evaluator from the National Agency for Quality Evaluation and Accreditation of Spain (ANECA). He is president from Engineering and Architecture Commission in Fundación Madrid+D. He is patron of the Spain Free Software Foundation. He is member of different committees at ISO Organization. He has published more than 200 research works in several relevant journal. He also manages different research projects in IoT, TEL, and Artificial Intelligence.