Cloud Environment Task Scheduling Optimization of Modified Genetic Algorithm

K. Malathi[1*], Dr.R. Anandan[2] and Dr.J. Frank Vijay[3]

[1*]Research Scholar, Department of Computer Science and Engineering, Vels Institute of Science Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India.
malathisaravanan2013@gmail.com

[2]Professor, Head of the Department, Department of Computer Science and Engineering, Vels Institute of Science Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India.
orcid.org/0000-0001-5461-1040

[3]Professor, Head of the Department, Department of Information and Technology, KCG College of Technology, Chennai, Tamil Nadu, India.

Received: September 17, 2022; Accepted: November 21, 2022; Published: February 28, 2023

Abstract

From the availability of resources to the accomplishment of tasks, cloud computing is a development of supercomputing. One of the most trustworthy paradigms in computing technology is built on internet-based parallel and distributed computing models. Optimization algorithms can be used to distribute user workloads to provided logical resources termed 'Virtual Machines' in the cloud computing system, which is a major aspect of resource management (VM). A fundamental challenge in cloud computing is the dynamic heterogeneity of resources and workloads, which necessitates efficient task scheduling and distribution. It is possible that task scheduling in distributed environments will improve our understanding of workflow scheduling, independent task scheduling that takes into account security and execution time for applications, trust between various system entities, and improved system utilisation and energy efficiency, among other things. The goal of this research is to contribute to these advancements in these areas: An independent task scheduling system based on genetics is presented to obtain the best outcomes in terms of time and resource consumption while allocating tasks to resources in accordance with the task's security needs. Various meta-heuristic algorithms, such as Genetic Algorithm, are currently being used to solve task scheduling difficulties.

Keywords: Cloud Computing, Modified Genetic Optimization, Virtual Machines, Expectation Maximization, Min-min, Max-min.

1 Introduction

Additionally, in cloud systems, the job scheduling process is hampered by the difficulty of load balancing. Scheduling tasks so that resources are not over or underutilised is a part of this procedure. As a result, the scheduling system must ensure that all of the resources are distributed in a proportional manner to their capabilities [1]. The problem isn't just overloaded VMs, but also idle VMs that aren't

Journal of Internet Services and Information Security (JISIS), volume: 13, number: 1 (February), pp. 34-43
DOI: *10.58346/JISIS.2023.I1.004*

*Corresponding author: Research Scholar, Department of Computer Science and Engineering, Vels Institute of Science Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India.

getting enough resources. VMs that are left idle for long periods of time hurt the service provider's bottom line. Every cloud computing framework needs to integrate task scheduling in order to manage the resources properly and efficiently serve cloud users. Assigning resources to incoming tasks that have performance optimization limitations is a critical function of this algorithmic mechanism [2]. Task scheduling is the only way to achieve high performance, high profit, high utilisation, scalability, provision efficiency, and cost-effectiveness in any cloud computing framework.

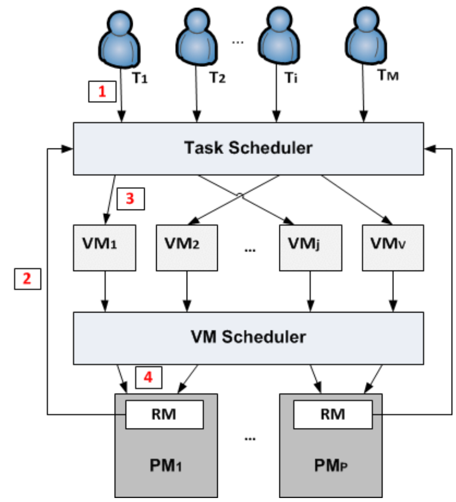

Figure 1: Cloud Computing-based Task Scheduling

Because the distribution of tasks is not always uniform, effective management and the ability to find the best available VM resources are required. When the cloud user needs to perform tasks, they can use the virtualized environment provided by a VM [3-7]. The resource provisioning principles allow a single server machine to be multiplexed into several VMs, each with its own operating system and configuration. Scheduling of tasks in the cloud is a critical topic. Popular cloud computing systems typically use local area networks (LANs) to connect their computing resources (some use Infiniband). Computing nodes in a virtualized cloud system can be standard PCs, servers, or even high-performance clusters, on which VMs can be created and deployed. There are a number of ways to divide a physical server into multiple virtual machines. The processing power of a virtual computer is the most fundamental unit of resource. Generally speaking, the resources a virtual machine has and the number of virtual machines a physical node may host are fixed. As an example, one CPU core is a virtual machine (VM) [8-11]. The goal of task scheduling is to find suitable virtual machines (VMs) for certain activities.

2 Background

A job might include a variety of duties. In this paper, all of the tasks are computational in nature, and they are not interdependent. In order to develop and execute an efficient scheduler, task scheduling is a major issue because it is an NP-complete problem [12]. These issues can be solved in two ways. Cupidity algorithm, branch and bound approach, and Dynamic Program are examples of exact solutions. Even while these algorithms are capable of finding an accurate solution, their efficiency plummets as the search space grows. We can use specific techniques to speed things up, but they don't actually improve the complexity of the problem and can't handle large-scale requirements [13]. As an alternative to scanning the full domain, heuristic algorithms are a useful option. A genetic algorithm, as is typical, exhibits impressive search capabilities.

A population-based optimization tool, the Genetic Algorithm was first introduced by Holland in 1975. Each chromosome in the population represents a possible solution to a problem and is formed of a string of genes in the GA model. For successful search and optimization, GA is one of the most extensively utilised artificial intelligence approaches. The advantage of this method is that it can handle a large search space, apply to a complex objective function, and avoid becoming stuck in a local optimal solution. Selection, genetic operation, and replacement make up a basic GA [14]. Genetic algorithm terminology includes steps like

Steps Involved in Genetic Algorithm

Step 1: Random Initialize a population

Step 2: Evaluate the fitness value of each population

Step 3: while either maximum number of iterations are exceeded or optimum solution is found Do:

- Initial Population: The GA uses the entire population of people to discover the best solution.
- Fitness Function: It is used to assess the quality of the population as a whole, based on a specific goal.
- Selection: One way to figure out how likely it is that different members of a population will pass on their genes to the next generation is by using the percentage selection operator.
- Crossover: For the individual code, a single point crossover operator is employed in just one intersection, at which time one of the pair's chromosomes is swapped.
- Mutation: To create a new individual, the values at a certain gene locus on a chromosomal coding series are replaced with those from other genes.

When it comes to load balancing, dynamic load balancing is preferable to static load balancing due to the fact that the cloud computing basis is totally reliant on real-time data. Depending on your requirements and preferences, you can use either task scheduling time or post-scheduling time for dynamic load balancing in your applications. In the first example, it is possible to allocate tasks based on the fluctuation of the workload on the VMs by simultaneously monitoring and researching the workload on the VMs as well as resource matching capabilities [15]. In the second case, after task scheduling has been completed, a load balancing method is carried out. The workload of a VM must be evenly divided across all of its virtual machines (VMs), or its limited duties must be moved to another VM. It is not uncommon for VM migration to cause delays, increased costs, and increased wait times. The ability to use load balancing to prevent unnecessary VM migrations during task scheduling is a fascinating and important tool to have available. Task scheduling is directed by examining the load on each virtual machine in the context of the current state of the system and making adjustments to decision variables as needed. Studies conducted in the past have looked into the usage of EA techniques to plan jobs in order to achieve load balancing.

3 Existing Algorithm

E-step and M-step are employed sequentially until convergence is achieved (no change in estimated parameter values). To begin, the (normalised) probability of assigning each data point to either model is calculated using the E-step. Then, the M-step is used to calculate the MLE for each model's mean and covariance matrix as a weighted average of the data points. This diagram shows the Expectation Maximization method in action:

Algorithm for Expectation Maximization

1: Start

Initialize inputs: $\sum$ and μ

Output $\{\widehat{\mu_k}\}\{\widehat{Z_k}\}$

2: E Step

Fix $\sum$ and μ

$$Z_k^i = \frac{g_k(X_i|\mu_{k,}\Sigma_k)}{\sum_{k=1}^{k} g_k(X_i|\mu_{k,}\Sigma_k)} \qquad (1)$$

3: Update Z_k^i

$$\widehat{\mu_k} = \frac{1}{Z_k}\sum_{i=1}^{N} Z_k^i X_i \qquad (2)$$

4: M step

Fix Z_k^i

$$\widehat{Z_k} = \frac{1}{Z_k}\sum_{i=1}^{N} Z_k^i (X_i - \widehat{\mu_k})\,(X_i - \widehat{\mu_k})^T \qquad (3)$$

5: Update $\sum$ and μ

$$Z_k = \sum_{i=1}^{N} Z_k^i \qquad (4)$$

6: If not converted go to E Step

7: If converted go to

$\{\widehat{\mu_k}\}\{\widehat{Z_k}\}$

8: Stop

4 Proposed Methodology

In this study, the researchers hope to devise a task scheduling technique that will allow for the quickest possible execution time while still dividing the burden evenly among all available resources [15]. This is a massive project on many levels. A combination of GA and Max-Min and Min-Min heuristic techniques has been used to achieve this goal, and they have been merged with GA in the proposed algorithm. The Main Controller and the Load Balancer are the two most significant cloud components for balancing load in order to achieve optimal performance. The Main Controller assigns the jobs to the relevant cloud divisions based on their nature. Each partition contains a Load Balancer, which periodically updates the status information and then dispatches the jobs depending on that information [16]. A diagram of the relationships between the Load Balancers, the Main Controller, and the nodes is shown in Figure 2Following the collection of status information from each node, the Load Balancers in each partition determine how the workload should be divided. When a piece of work is delivered to the Public Cloud, its cloud partition statuses are categorised into three categories: idle, normal, and overloaded. The first step is to choose the most appropriate division for your needs.

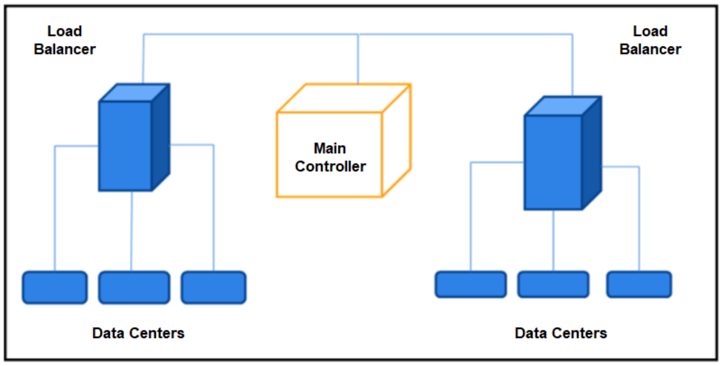

Figure 2: Controller based Load Balancing Working Architecture

On arrival, Main Controller requests cloud partition for information on jobs' location [17]. The task is performed locally, unless a partition that isn't overloaded is discovered, in which case it is handled remotely if the location state is idle or normal.

As seen in Figure 3, the three fundamental events in a genetic algorithm are the establishment of an initial population, a crossover, and a mutation [18]. Similar to the human genetic process, there is an increase in performance with each successive generation. A fitness function is built in accordance with the proposed method in order to select the genetically better population and to remove the inferior outcomes in later rounds.

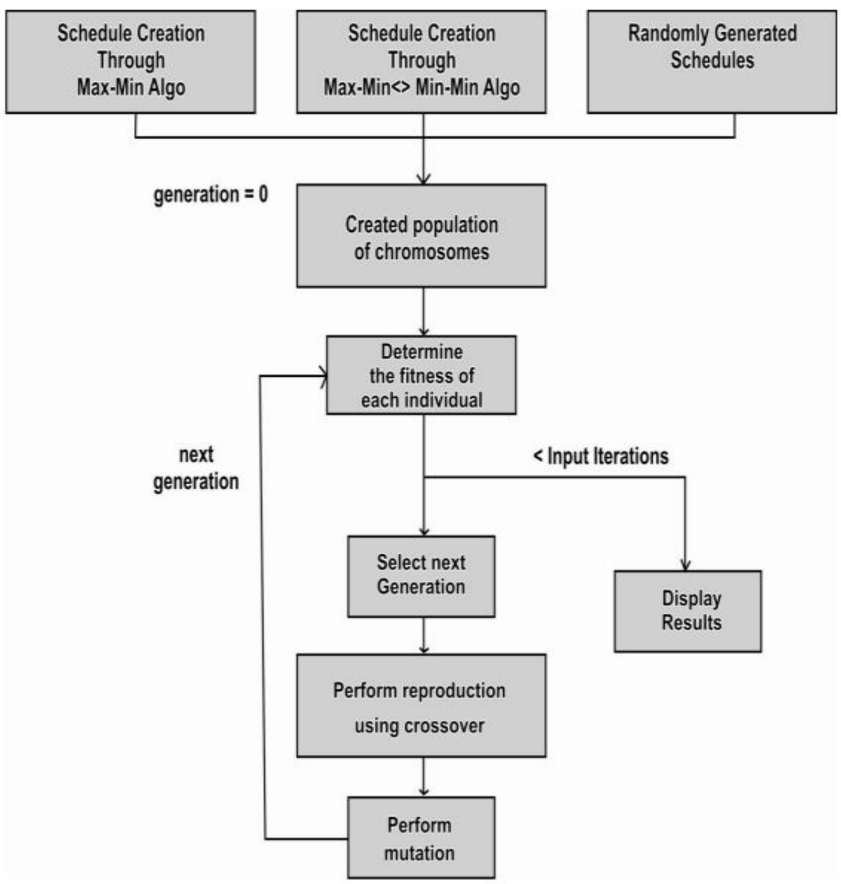

Figure 3: Flowchart for Modified Genetic Algorithm

Algorithm 1: Modified Genetic Algorithm

1: Start
2: Input → Generate an initial population do until termination criteria
3: Step 1: Calculate the task to resource mapping
4: Step 2: Evaluate

$$f(x_{mi}) = \frac{m_i \, X \, (1-s_i)}{m \, X \, e^{-\frac{1}{r}}} \quad (5)$$

m_i= Makespan
m= Largest makespan in iteration
$s_i = \frac{Successfull\ no.of\ Completions}{Total\ no.of\ Jobs}$
r= Reliability parameter

5: Step 3: Probabilistically choose the best solution
6: Step 4: Check for new optimal solution
7: Step 5: update best solution found so far and generate new population based on the best solution reached
8: Stop

5 Simulation Results

Experiments and results generated for the suggested and traditional algorithms are described in this section. The analysis of these results follows. For the purpose of testing and comparing the performance of the Genetic Algorithm with the Standard Genetic Algorithm, a simulation programme written in MATLAB and running on an Intel Core i3 3.70 GHz processor with 4 GB RAM has been developed. A summary of the parameters utilised in the simulation is provided in Table 1.

No.of Tasks	Min-Min	Max-Min	Existing EA	Existing GA	Proposed GA
4	46	43	45	42.4	41.5
8	93.3	87.5	90.2	86.2	83.6
16	180.1	176.6	178.3	175.3	169.4
32	372.8	346.4	359.8	345.8	345.4
54	758.5	708.5	732.3	706.4	689

Table 1: Parameters for Simulation

Parameter	Values
No. of Machines	10
No. of Tasks	10-60
Crossover Operations	Uniform-point Crossover
No. of Iterations	150
Condition for Termination	No. of Iterations
Mutation Operator	Rebalancing

Other techniques of calculating probability may exist, but fitness must be considered. Equation below determines the likelihood of each food source based on its fitness.

$$P_{ij} = \frac{F(X_{mi})}{\sum_{i=1}^{SN} F(X_{mi})} \quad (6)$$

Existing and proposed GA algorithms are shown in Table 2 by their Makespan values. In comparison to conventional approaches such as Min-Min and Max-Min, and the usual Genetic method, the Makespan has been observed to be reduced utilising the Genetic method. It takes 43.9 seconds longer to schedule 40 jobs using the Genetic technique (as shown in Table 2) than it does to use the Proposed Genetic Algorithm (as shown in Table 2).

Table 2: Makespan Values obtained for Existing and Genetic Algorithm and Proposed Genetic Algorithm

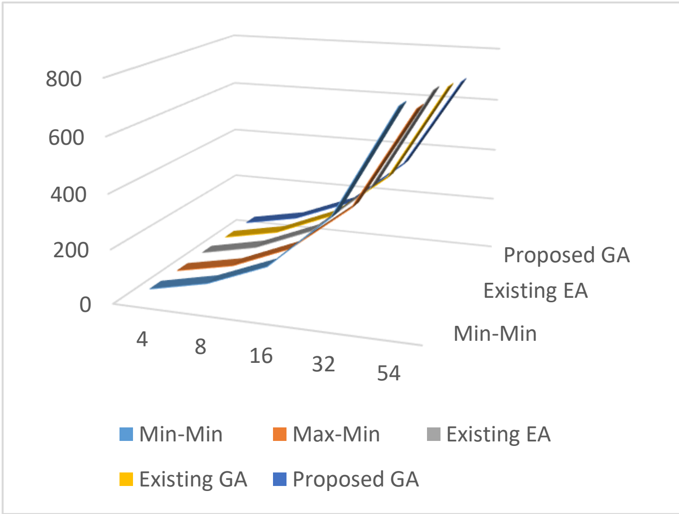

Figure 4: Plot for Makespan Values Obtained for Existing and Genetic Algorithm and Proposed Genetic Algorithm

Figure 4 disputes the Plot for Makespan Values obtained for Existing and Genetic Algorithm and Proposed Genetic Algorithm.

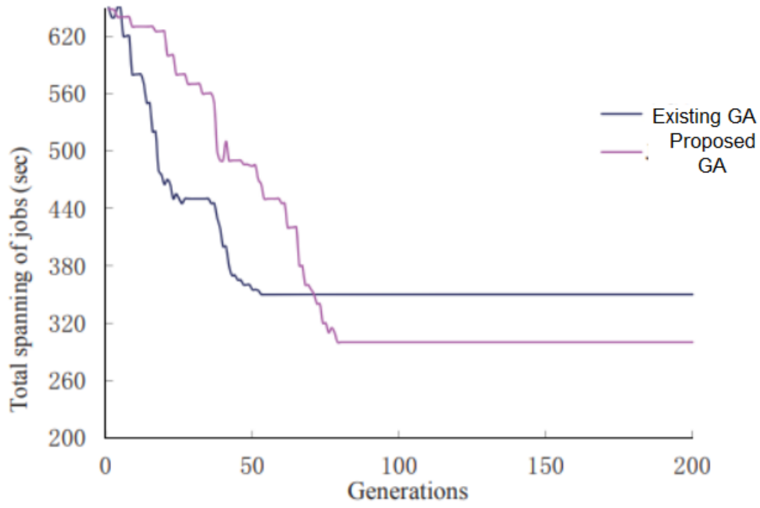

Figure 5: Total Job Running Time

Figure 5 represents the comparison of the both Proposed and existing Genetic Algorithms of runtime.

6 Conclusion

An initial proposal for a load balancing model was made during the preliminary stages of the investigation. According to the findings of the research, implementing a partitioning strategy can reduce the amount of time required to administer a cloud system. According to the findings of the proposed research, the management overhead of the cloud environment can be reduced by partitioning the cloud environment. Cloud partitioning is accomplished by the use of coordinate values, which solves the problem of cloud division rules. With the help of efficient job scheduling, cloud service providers may better meet the needs of their clients and increase their revenue. The deployment of load balancing has become an extra significant task as a result of the increasing increase in client demand. In order to optimise performance, the output of the task scheduler should take into account the distribution of workload. The researchers completed their investigation by employing a Genetic Algorithm-enhanced load-balanced meta-heuristic scheduling strategy that was load balanced. It has been recommended in the current research that a multi-objective fitness function be used in order to conduct out specialised investigations on load-balanced meta-heuristic algorithmic methods. An effective load balancing algorithm, which aids in the realisation of Green Computing, can be investigated in the future to ensure that the load is maintained in an even distribution.

Declarations:

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Availability of Supporting Data:

Datasets derived from public resources and made available in the article.

Competing Interests: The authors declare that there are no relevant financial or non-financial competing interests to report.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Funding Statement: The **author** received no specific funding for this study.

Author's Contributions

First author K. Malathi has made substantial contributions to conception and design, or acquisition of data, or analysis and interpretation of data and has been involved in drafting the manuscript and revising it critically for important intellectual content. Second author R. Anand has given final approval of the version to be published. And Authors agree to be accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

Acknowledgements: I would like to express my special thanks of gratitude to my husband who helped me to complete the project within the time frame.

References

[1] Yao, F., Pu, C., & Zhang, Z. (2021). Task duplication-based scheduling algorithm for budget-constrained workflows in cloud computing. *IEEE Access*, *9*, 37262-37272.

[2] Marahatta, A., Pirbhulal, S., Zhang, F., Parizi, R.M., Choo, K.K.R., & Liu, Z. (2019). Classification-based and energy-efficient dynamic task scheduling scheme for virtualized cloud data center. *IEEE Transactions on Cloud Computing*, *9*(4), 1376-1390.

[3] Zhu, L., Huang, K., Hu, Y., & Tai, X. (2021). A Self-Adapting Task Scheduling Algorithm for Container Cloud Using Learning Automata. *IEEE Access*, *9*, 81236-81252.

[4] Sahoo, S., Sahoo, B., & Turuk, A.K. (2019). A learning automata-based scheduling for deadline sensitive task in the cloud. *IEEE Transactions on Services Computing*, *14*(6), 1662-1674.

[5] Alsadie, D. (2021). A metaheuristic framework for dynamic virtual machine allocation with optimized task scheduling in cloud data centers. *IEEE Access*, *9*, 74218-74233.

[6] Chen, L., Guo, K., Fan, G., Wang, C., & Song, S. (2020). Resource constrained profit optimization method for task scheduling in edge cloud. *IEEE Access*, *8*, 118638-118652.

[7] Mishra, S.K., Mishra, S., Alsayat, A., Jhanjhi, N.Z., Humayun, M., Sahoo, K.S., & Luhach, A.K. (2020). Energy-aware task allocation for multi-cloud networks. *IEEE Access*, *8*, 178825-178834.

[8] Chen, X., Cheng, L., Liu, C., Liu, Q., Liu, J., Mao, Y., & Murphy, J. (2020). A WOA-based optimization approach for task scheduling in cloud computing systems. *IEEE Systems Journal*, *14*(3), 3117-3128.

[9] Yuan, H., Bi, J., & Zhou, M. (2019). Multiqueue scheduling of heterogeneous tasks with bounded response time in hybrid green IaaS clouds. *IEEE Transactions on Industrial Informatics*, *15*(10), 5404-5412.

[10] Geng, S., Wu, D., Wang, P., & Cai, X. (2020). Many-objective cloud task scheduling. *IEEE Access*, *8*, 79079-79088.

[11] Chen, Z., Hu, J., Chen, X., Hu, J., Zheng, X., & Min, G. (2020). Computation offloading and task scheduling for DNN-based applications in cloud-edge computing. *IEEE Access*, *8*, 115537-115547.

[12] Xiong, Y., Huang, S., Wu, M., She, J., & Jiang, K. (2017). A Johnson's-rule-based genetic algorithm for two-stage-task scheduling problem in data-centers of cloud computing. *IEEE Transactions on Cloud Computing*, *7*(3), 597-610.

[13] Al-Maytami, B.A., Fan, P., Hussain, A., Baker, T., & Liatsis, P. (2019). A task scheduling algorithm with improved makespan based on prediction of tasks computation time algorithm for cloud computing. *IEEE Access*, *7*, 160916-160926.

[14] Xu, J., Hao, Z., Zhang, R., & Sun, X. (2019). A method based on the combination of laxity and ant colony system for cloud-fog task scheduling. *IEEE Access*, *7*, 116218-116226.

[15] Pang, S., Li, W., He, H., Shan, Z., & Wang, X. (2019). An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing. *IEEE Access*, *7*, 146379-146389.

[16] Hu, K., Zeng, G., Ding, S., & Jiang, H. (2019). Cluster-scheduling big graph traversal task for parallel processing in heterogeneous cloud based on dag transformation. *IEEE Access*, *7*, 77070-77082.

[17] Yuan, H., Bi, J., & Zhou, M. (2019). Multiqueue scheduling of heterogeneous tasks with bounded response time in hybrid green IaaS clouds. *IEEE Transactions on Industrial Informatics*, *15*(10), 5404-5412.

[18] Guo, P., Liu, M., Wu, J., Xue, Z., & He, X. (2018). Energy-efficient fault-tolerant scheduling algorithm for real-time tasks in cloud-based 5G networks. *IEEE Access*, *6*, 53671-53683.

[19] Boualem, A., De Runz, C., & Ayaida, M. (2022). Partial Paving Strategy: Application to optimize the Area Coverage Problem in Mobile Wireless Sensor Networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, *13*(2), 1-22.

Authors' Information

K. Malathi is currently doing Ph.D in Computer Science and Engineering at the Vels Institute of Science, Technology& Advanced Studies -VISTAS, Chennai, Tamil Nadu. She started her career as Lecturer in J.J College of Engineering and Technology Tiruchirappalli. She has published more than five papers and chapters. Later briefly working in different technology-based research roles, her interests lie at the field of cloud computing and Internet of things and also had worked as a Teaching faculty in IGCSE, Malaysia. She had attended so many Conferences, seminar and Faculty development Programmes.

Dr.R. Anandan is currently working as Professor and Head, Department of Computer Science and Engineering, School of Engineering, VelsInstitute of Science, Technology & Advanced Studies (VISTAS), Chennai, Tamil Nadu, India which is Pioneer Institution in Engineering. He has vast experience in corporate and all levels of Academic in Computer Science and Engineering his knowledge of interests not limited to Artificial intelligence Soft Computing, Machine Learning, High Performance Computing, Big data Analytics, Image Processing, 3D printing and knowledge Engineering. He has published more than 120 research papers in various International.