International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

TASK SCHEDULING IN FOG COMPUTING ENVIRONMENT: AN OVERVIEW

***Nikita Sehgal[1], Savina Bansal[2] and RK Bansal[3]**
[1,2,3]Department of Electronics & Communication Engineering
[1,2,3]Giani Zail Singh Campus College of Engineering & Technology,
Maharaja Ranjit Singh Punjab Technical University
Bathinda-151001 (Punjab), India
https://orcid.org/0000-0002-7975-4061

Abstract: *The number of Internet of Things (IoT) devices and sensors has significantly expanded in recent years. In general, fog computing increases cloud-based processing, storage, and networking capabilities by being located closer to IoT devices and sensors. Task scheduling is used to finish work in a set amount of time using a finite number of resources. The completion of tasks within the allotted time in fog computing is a key difficulty due to the increased amount of data that needs to be processed. Additionally, in order to identify current research gaps in the field of fog, we map the existing works to the taxonomy. This article offers a broad overview of various task and resource scheduling techniques used in fog computing. It examines and contrasts several techniques created for a fog computing environment to ascertain their contributions and limitations. Moreover, it offers encouraging study directions for other researchers working in this area.*
Keywords: *fog computing, scheduling, deadline, security*

1. Introduction

The Internet of Things (IoT) is one of the spotlight innovations which have the potential to provide unlimited benefits to our society. With the Internet of Things (IoT) becoming part of our daily life and our environment, we expect speedy growth in the number of connected devices. IoT is expected to connect billions of gadgets and humans to bring promising benefits for us. With this growth, fog computing, along with its related edge computing are seen as promising solutions for handling the large volume of security-critical and time-sensitive data that is being produced by the IoT. A distributed computing model known as "fog computing" serves as a middle layer between cloud datacenters and Internet of Things (IoT) devices and sensors. It provides computing and networking storage facilities in order to bring Cloud-based services closer to IoT devices [1]. Cisco first proposed the idea of fog computing in 2012 to overcome the difficulties IoT applications faced with traditional cloud computing. At the network's edge, IoT devices and sensors are widely dispersed together with real-time and latency-sensitive service requirements [5]. Because they are geographically centralized, cloud datacenters frequently struggle to meet the storage and processing needs of billions of geographically dispersed IoT devices and sensors. As a result, there is network congestion, excessive service delivery delay, and poor Quality of Service (QoS) [9].The comparison of various computing paradigms is summarized in Table 1. In this chapter, we elaborately discuss the key differences of Fog computing with other computing paradigms.

The rest of the paper is organized as follows: scheduling objectives are presented in Section 2. Section 3 discusses various task scheduling strategies. Challenges and research gaps are given in Section 4. Finally, conclusions are presented in Section 5.

Table1: Comparison on different computing paradigms

Attributes	Cloud Computing	Edge Computing	Fog Computing
Architecture	Centralized	Distributed	Distributed
Execution Time	High	Medium	Low
location	No	Yes	Yes

DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

Main Providers	Amazon and Google	Cellular network providers	Proprietary
Mobility	Inadequate	Offered with limited support	Supported
Interaction in Real-Time	Available	Available	Available
Security	Centralized (guaranteed by the Cloud provider)	Centralized (guaranteed by the Cellular operator)	Mixed (depending on the implementation)
Energy Consumption	High	Low	Varying but higher than for edge
Bandwidth Cost High	High	Low	Low High
Storage Capacity& Computation High	High	Very limited	Varying High
Scalability	Average	High	High

2. Scheduling Objectives

The design of the scheduling algorithm is highly dependent upon the type of the system. In other words, the scheduling algorithm which is better for a system may not be good for the other system having different conditions. Depending upon the system, the user and the designer might expect the following from a scheduler.

- *Maximum throughput:* It means that maximum number of processes should be completed per unit time.
- *Priorities application:* If the system is priority-based means priorities are assigned to different tasks of the system then the scheduler should follow the highest priority first.
- *Minimum overhead:* Overhead causes wastage of resources. So, scheduler is required which makes best possible use of the equipment's that are available for the use.
- *Reduction in waiting time:* There should be no waiting in the process before or while execution.
- *Reduction in rejection ratio:* The scheduler must take care that all the jobs meet their deadline.
- *Utilization of resources:* All the resources should be utilized in a proper manner to get best results.
- *Decency:* The scheduler must take care that each process gets its fair share of CPU and no one suffer heavy blocking.

3. Task Scheduling Strategies

The goal of task scheduling is to distribute a set of jobs among fog nodes in order to satisfy QoS criteria while minimizing task execution and transmission times [19,20].Task scheduling algorithms can be classified as static or dynamic algorithms and these are briefly discussed in the following subsections.

3.1 Static Task Scheduling Algorithms

In static scheduling, all details regarding the jobs and resources in the fog should be accessible. Heuristic-based and metaheuristic-based algorithms are two subcategories of static scheduling algorithms. Famous algorithms including the genetic algorithm, ant colony optimization, bee life, search annealing, and symbiotic organization have been used in metaheuristic scheduling to enhance the scheduling process [21]. In the cloud-fog environment, Xu et al. [36] presented an associated work scheduling technique based on Laxity-Based Priority and Ant Colony System (LBP-ACS). There are two parts to the suggested strategy: Constrained Optimization Algorithm based on the ACS

International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 1 Volume No.7 January - February – 2023
DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

and LBP Algorithm (LBPA) (COA-ACS). The task priority order is obtained using the first part (LBPA), and the task scheduling scheme is obtained using the second portion (COA-ACS). The experimental results demonstrate that the proposed algorithm when compared with Greedy for Energy (GfE), Heterogeneous Earliest Finish Time (HEFT) [11], and Hybrid Ant Colony Optimization with Differential Evolution Algorithms [12] can reduce the energy consumption of processing all tasks and reduce the failure rate of associated tasks scheduling with mixed deadlines by using the CloudSim simulation .

Hybrid-Earliest Deadline First (Hybrid-EDF) is a heuristic that Stavrinides and Karatza [10] suggested for the dynamic scheduling of real-time IoT operations in a three-tiered architecture. This approach aims to schedule computationally demanding jobs with low communication requirements in the cloud and communication-intensive operations with low computational requirements in the fog. The task selection phase and the virtual machine (VM) selection phase are the two steps of the suggested scheduling technique. The task's priority is determined by its earliest deadline. The task with the higher average computational cost is chosen first when there are two or more jobs with the same priority. The work is chosen by the scheduler and sent to the VM that can do it as soon as possible. A four-tier design for delay-aware scheduling and load balancing in the fog environment is suggested by Sharma and Saini [27]. iFogSim simulates the proposed algorithm, which produces superior outcomes when compared to nine techniques: Graph partitioning [28], Simple scheduling [30], Delay Energy Balanced Task Scheduling (DEBTS) [40], Delay Energy Balanced Task Scheduling (CMaS) [18], BLA [21], NSGA-II [29], HEFT [11], Multi-Population Genetic Algorithm (MPGA) [34], and Dynamic Resource Allocation Method (DRAM) [37] in terms of response time, scheduling time, load balancing rate, latency, and energy use. Although it can further minimize delay and total dependence, the suggested algorithm does not address execution costs, and the authors do not address data replication techniques for maintaining data in a fog computing network.

To address the issue of terminal devices with constrained computational capabilities and high energy consumption, Wang and Li [34] offer a job scheduling technique based on a Hybrid Heuristic (HH) algorithm for various fog nodes. The Improved Particle Swarm Optimization (IPSO) method and the Improved Ant Colony Optimization (IACO) algorithm [13] are combined in the HH algorithm. MATLAB is used by the writers to evaluate their work. Results of the experiment demonstrate that the HH algorithm outperforms IPSO, IACO, and Round Robin (RR) on three performance criteria (Make-span, energy consumption, and reliability). Task clustering and fog node clustering are not applicable to this approach. A task scheduler is suggested by Pham et al. [38] to boost the effectiveness of large-scale offloading applications. A suitable trade-off between make span and cost is desired. The scheduler arranges the jobs according to how long the critical pathways are between each task and the last task before choosing the best node to carry out each task. The authors take into account that the fog provider rents virtual hosts and network bandwidth from cloud providers to increase the functionality of the fog nodes when figuring out the financial cost. The suggested method creates a good trade-off between the makespan and the cost of task execution, according to the results.

3.2 Dynamic Task Scheduling Algorithms

The scheduling process is dynamic when task priorities are determined while the system is running. Since requests come in real-time, there is no prior knowledge of the resources required for the task in dynamic scheduling. Real-time scheduling approaches and heuristic scheduling approaches can both use dynamic scheduling methods [37]. In the real-time scheduling method, a job will already be on a machine when it comes. However, before scheduling, the tasks are gathered into a collection of ideal solutions in the heuristic scheduling strategy. Subbaraj and Thiyagarajan [33] had suggested a methodology for performance-oriented task-resources mapping in a fog computing environment. To analyze the technical specifications of the fog devices, two distinct multi-criteria decision-making processes are used. In the first technique, the Analytic Hierarchy Process (AHP) was used to compute priority weights and rank fog devices. The second technique uses AHP to create priority weights, and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) algorithm to order the fog devices based on the weights calculated by AHP. Shahid, et al. [30] had proposed a popularitybased caching strategy in content delivery fog networks. Two energy-conscious approaches, content screening and load balancing were applied in this case. In the proposed approach, popular components are determined by random distribution and classified into three types. The suggested techniqueconsumes92.6 percentand82.7 percent less energy than no caching and basic caching systems, respectively

3.2.1 Energy based Scheduling

Bansal et al.[29] used dynamic voltage scaling (DVS) and dynamic power management (DPM) techniques for energy management while scheduling preference-oriented fixed-priority periodic real-time tasks. Preference-oriented energy-aware rate-monotonic scheduling (PER) and preference-oriented extended energy-aware rate-monotonic scheduling (PEER) algorithms were proposed that maximize energy savings while fulfilling preference-value of tasks. Extensive simulations show that PER and PEER outperforms in terms of energy savings when compared to several related studies. Azizi, et al. [3] had mathematically modeled the task scheduling problem to reduce total fog node (FN) energy consumption while achieving IoT task quality of service (QoS) standards. Priority-aware semi-greedy (PSG) and PSG with multistate process (PSG-M), two semi-greedy-based methods, were devised to successfully map IoT tasks to FNs. The suggested solutions enhance the percentage of jobs that meet their deadline requirement by 1.35x while decreasing overall deadline violation time by 97.6%. Fellir et al.[8] suggested the fog computing paradigm to boost efficiency and address other issues in IoTdatastorageand processing by providing IoT data processing and storage functionality locally at the edges (IoT devices) rather than sending all data to the centralized cloud. A multi-agent-based model was proposed in this paper to evaluate task scheduling on a cloud-fog computing platform, with the goal of serving the most important work first, taking into account job priority, wait time, status, and the resources required to complete it successfully. The simulation results of the proposed model show that it can increase resource consumption and performance. Hosseinioun et al. [12] proposed an energy-conscious approach to energy usage based on the dynamic voltage and frequency scaling (DVFS) methodology [24]. A hybrid Invasive Weed Optimization and Culture(IWO-CA)evolutionary method was also used to generate acceptable task sequences. The recommended algorithm improves on several present algorithms in terms of energy usage, according to the experimental results of the suggested study.

3.2.2 Security based Scheduling

Razaque, et al. [22] had suggested an energy-efficient and safe method for the mobile fog-based cloud to help IoT to minimize energy usage. The voltage scaling factor is used by the EESH algorithm to minimize energy usage.When the number of IoT activities increases, the EESH algorithm outperforms state-of-the-art competitive algorithms, and EESH consumes extremely little energy. Furthermore, because identifying mobile cloud users was critical, the EESH was fortified by a fake data detection system based on block chain technology. Javanmardi et al. [15] proposed security-aware task scheduler algorithm tailored for IoT-fog networks called FUPE for SDN-based fog networks. SDN switches are powerful devices that may function as fog devices as well as fog gateways. As a result, fog devices are more vulnerable to a range of dangers. FUPE is based on fuzzy logic that combines optimal computing resources and suitable levels of data safety into a single synthetic goal to obtain a single correct response. In this scenario, several optimization approaches such as MOPSO and evolutionary algorithms are employed to combat other sorts of assaults in IoT fog networks.

Singh, et al. [32] have suggested a RT-SANE (Real-Time Security Aware Scheduling on the Network Edge), which allows batch and interactive applications to run while keeping deadlines and safety requirements in mind. RT-SANE chooses among mdc and acdc based on network latency and security tags. In the proposed work, RT-SANE employs a distributed orchestration design and interface that prioritizes speed and reliability. But the suggested work not schedules the task in the energy efficient manner. An improved duplication strategy for arbitrary task graphs with a limited number of interconnection-constrained processors is proposed by Bansal [25]. Unlike most other algorithms, which replicate all possible parents/ancestors of a given task, the proposed algorithm avoids

DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

redundant duplications and only duplicates nodes if they help improve performance. This reduces duplications as well as time and space complexity. A variety of parallel numerical applications are represented by simulation results for a clique and an interconnection-constrained network topology with random and regular benchmark task graph suites[26]. Performance is compared in terms of normalized schedule length and efficiency to some well-known and recently proposed algorithms.

3.3 Resource Scheduling

The resource allocation strategy and computation of loading were suggested by Du et al. [7] and were initially taken into consideration for loading decision-making. They suggested a low-complexity technique to address the optimization problem, where the optimization decisions were obtained using randomization and semi-definite relaxation. By using fractional programming theory and Lagrangian dual decomposition [4], the resource allocation was achieved. The proposed model performed better than other mentioned classification algorithms, optimized transmit power and bandwidth, and had a slower response time. Liu et al. [17] proposed the queuing theory for a thorough investigation of the cost of compute loading, response time, and energy consumption in a fog computing system. A multi-objective optimization challenge was developed and formulated with a shared goal of decreasing cost, response time, and energy consumption while determining the ideal location for transferring power and loading probability for each device. The main flaw of this article is that the researchers made no mention of resource allocation, availability, or security in this architecture.

Shah-Mansouri and Wong [31] suggested a resource allocation method for fog-based IoT systems. In order to increase the performance quality of applications, each user tried to maximize the computation offloading choice for each job that was received by their devices. The paper, however, was not designed as an online resource allocation technique, and thus neglected to take the dynamic arrival of compute tasks into account. Wang and Chen [35] created a resource allocation technique and a loading decision for IoT networks in a fog environment that decreased latency. They created a combined optimization problem with the offloading choice, local computing power, and computing resource allocation for a fog node as the parameters. However, the authors did not examine the model of multiuser multitasking or evaluate the usability and efficiency of research plans in real-world applications

4. **Issues And Challenges**

Several task scheduling methods have been described above in the fog computing environment. Different researchers focus on different parameters for improving service quality, networks performance, power consumption, etc. In fog and cloud computing, some of them also emphasized makespan, workload balancing, financial cost, response time, computing resource utilization, and effective energy utilization. Some researchers validated their approaches through simulation, while others did not test them experimentally. To overcome the limitations, attention should be paid to the following issues, particularly in the fog computing environment.

- There is a scarcity of infrastructure for validating a true fog computing environment. Therefore, much fog computing research has been validated using simulation tools. Therefore, there is a need to develop a large - scale and real - time testbed(s) for validating fog computing approaches
- Fog computing policies such as task scheduling and resource scheduling must be developed taking into consideration the heterogeneity of fog computing devices
- Energy and Security aspect should be considered while scheduling the tasks.

5. **Conclusions**

The primary obstacles against the development are various security threats and huge energy consumption [2]. As per the complexity of the modern world, it is required that energy consumption must be minimized while the execution of the job is taking place. Although energy aware task scheduling schemes have been independently studied widely, but the major problem in the Fog

DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

computing is security due to the limited resources [14]. There is a need to identify the characteristics of security in Fog environment [16].

References

[1]A.Dastjerdi, H. Gupta, R. Calheiros, S. Ghosh, and R. Buyya (2016), Chapter 4—fog computing: Principles, architectures, and applications, In Internet of Things: Principles and Paradigms, ed. R. Buyya, and A.V. Dastjerdi, New York: Morgan Kaufmann, pp. 61-75.

[2]Arwa Alrawais, Abdulrahman Alhothaily, Chunqiang Hu and Xiuzhen Cheng (2017),Fog Computing for the Internet of Things: Security and Privacy Issues, IEEE Internet Computing, 21(2), pp.34-42, DOI: 10.1109/MIC.2017.37

[3]Azizi, S., Shojafar, M., Abawajy, J., &Buyya, R. (2022). Deadline-aware and energy-efficient IoT task scheduling in fog computing systems: A semi-greedy approach. Journal of Network and Computer Applications, 103333.

[4]Bal, P. K., Mohapatra, S. K., Das, T. K., Srinivasan, K., & Hu, Y. C. (2022). A Joint Resource Allocation, Security with Efficient Task Scheduling in Cloud Computing Using Hybrid Machine Learning Techniques. Sensors, 22(3), 1242.

[5]Bonomi, F., Milito, R., Zhu, J., Addepalli, S.(2012), Fog computing and its role in the internet of things. Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing - MCC'12, p. 13. doi.org/10.1145/2342509.2342513.

[6]C.K.Chen, Y.H.Chang, Y.T.Chen ,C.C.Yang, J.K.Lee, (2007) Switching supports for stateful object remoting on network processors, Journal of Supercomputer 40, pp.281–298.

[7]Du J, Zhao L, Feng J, Chu X (2018) Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee, IEEE Transactions on Communications, 66(4), pp. 1594–1608.

[8]F Fellir, A El Attar, K Nafil,& L Chung,(2020). A multi-Agent based model for task scheduling in cloud-fog computing platform. In 2020 IEEE international conference on informatics, IoT, and enabling technologies (ICIoT), pp. 377-382.

[9]Fog computing and the internet of things: Extend the cloud to where the things are, cisco white paper. 2015. http://www.cisco.com/c/dam/en_us/solutions/trends/iot/ docs/computing-overview.pdf

[10] G.L. Stavrinides, H.D. Karatza (2019), A hybrid approach to scheduling real-time IoT workflows in fog and cloud environments, Multimedia Tools Applications 78, pp. 24639–24655.

[11] H. Topcuoglu, S. Hariri, M.y. Wu (2002), Performance-effective and low-complexity task scheduling for heterogeneous computing, IEEE Transactions Parallel Distributed Systems, 13, pp. 260–274.

[12] P Hosseinioun, M Kheirabadi, S. R. K. Tabbakh, & R Ghaemi, (2020), A new energy-aware tasks scheduling approach in fog computing using hybrid meta-heuristic algorithm, Journal of Parallel and Distributed Computing, 143, pp. 88-96.

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

[14] Jasleen Kaur, Alka Agrawal and Raees Ahmad Khan (2020), Security Issues in Fog Environment: A Systematic Literature Review, Springer- International Journal of Wireless Information Networks, 27, pp. 467-483, DOI: https://doi.org/10.1007/s10776-020-00491-7.

[15] S.Javanmardi, M Shojafar, R Mohammadi, A Nazari, V Persico, & A Pescapè (2021), FUPE: A security driven task scheduling approach for SDN-based IoT–Fog networks, Journal of Information Security and Applications, 60, 102853.

[16] Jimoh Yakubu, Shafi Muhammad Abdulhamid, Haruna Atabo Christopher, Haruna Chiroma and Mohammed Abdullahi (2019), Security Challenges In Fog-Computing Environment: A Systematic Appraisal Of Current Developments, Springer- Journal of Reliable Intelligent Environments, 5, pp. 209-233, DOI:https://doi.org/10.1007/s40860-019-00081-2.

[17] Liu L, Chang Z, Guo X, Mao S, Ristaniemi T (2017), Multiobjective optimization for computation ofoading in fog computing, IEEE Internet Things Jounal, 5(1), 283–294.

DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

[18] Medara, R., & Singh, R. S. (2021). Energy efficient and reliability aware workflow task scheduling in cloud environment. Wireless Personal Communications, 119(2), 1301-1320.

[19] Nirmal Kaur, Savina Bansal and Rakesh Bansal (2013), Energy aware scheduling strategies for distributed computing systems, International journal of advanced research in computer science and software engineering, 3, pp. 280-283

[20] N Kaur, S Bansal and RK Bansal (2016), Energy efficient duplication-based scheduling for precedence constrained tasks on heterogeneous computing cluster, Multiagent and Grid Systems 12 (3), pp. 239-252

[21] N Kaur, S Bansal, RK Bansal (2021), Survey on energy efficient scheduling techniques on cloud computing, Multiagent and Grid Systems, 17 (4), pp. 351-366

[22] Razaque, A., Jararweh, Y., Alotaibi, B., Alotaibi, M., Hariri, S., &Almiani, M. (2022), Energy-efficient and secure mobile fog-based cloud for the Internet of Things. Future Generation Computer Systems, 127, pp.1-13.

[23] S. Bitam, S. Zeadally, A. Mellouk (2018), Fog computing job scheduling - optimization based on bees swarm, Enterprise Information Systems ,12, pp. 373–397.

[24] S Bansal, P Kumar, K Singh (2002), Duplication-based scheduling algorithm for interconnection-constrained distributed memory machines, International Conference on High-Performance Computing, pp. 52-62

[25] S.Sarka, and S. Misra (2016), Theoretical modelling of fog computing: A green computing paradigm to support IoT applications, IET Networks,5(2), pp. 23-29

[26] Savina Bansal, Padam Kumar and Kuldip Singh (2003), An Improved Duplication Strategy for Scheduling Precedence Constrained Graphs in Multiprocessor Systems, IEEE Transactions on Parallel and Distributed Systems-TPDS, vol. 14, pp. 533-544.

[27] S. Sharma, H. Saini (2019), A novel four-tier architecture for delay aware scheduling and load balancing in fog environment, Sustainable Computing: Informatics and Systems, DOI:10.1016/j.suscom.2019.100355.

[28] S. Ningning, G. Chao, A. Xingshuo, Z. Qiang (2016), Fog computing dynamic load balancing mechanism based on graph repartioning,China Communications.13, pp.156–164.

[29] Savina Bansal, Rakesh Kumar Bansal, Kiran Arora (2020), Energy-cognizant scheduling for preference-oriented fixed-priority real-time tasks, Journal of Systems Architecture, vol. 108, doi.org/10.1016/j.sysarc.2020.101743

[30] M H Shahid, A R Hameed, S ul Islam, H A Khattak, I U Din, I. U., & J J Rodrigues, (2020), Energy and delay efficient fog computing using caching mechanism. Computer Communications, 154, 534-541.

[31] H Shah-Mansouri, VW Wong (2018), Hierarchical fog-cloud computing for IoT systems: A computation offloading game. IEEE Internet Things Journal, 5(4), pp. 3246-3257

[32] A Singh, N Auluck, O Rana, & S Nepal (2021), Scheduling Real Time Security Aware Tasks in Fog Networks, IEEE World Congress on Services.

[33] S Subbaraj & R Thiyagarajan, (2021), Performance oriented task-resource mapping and scheduling in fog computing environment, Cognitive Systems Research, 70, pp.40-50.

[34] T. Wang, Z. Liu, Y. Chen, Y. Xu, X. Dai (2014), Load balancing task scheduling based on genetic algorithm in cloud computing, Proceedings of the 2014 IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, IEEE, Dalian,China,pp.146–152.

[35] Wang Q, Chen S (2020), Latency-minimum offloading decision and resource allocation for fog-enabled internet of things networks, Transactions on Emerging Telecommunications Technologies,31(12), pp 1-14.

[36] X. Xu, S. Fu, Q. Cai, W. Tian, W. Liu, W. Dou, (2018), Dynamic resource allocation for load balancing in fog environment, pp. 1-15.

[37] X. Zhang, H. Duan, J. Jin (2008), DEACO: hybrid ant colony optimization with differential evolution, Proceedings of the 2008 IEEE Congress on Evolutionary Computation, IEEE World Congresson Computational Intelligence, Hong Kong, China, pp.921–927.

DOI:10.46647/ijetms.2023.v07i01.009 ISSN: 2581-4621

[38] Xuan-Qui Pham, Nguyen Doan Man, Nguyen Dao Tan TriN.D.T.Tri, N.Q.Thai, E.N.Huh (2017), A cost-and performance-effective approach for task scheduling based on collaboration between cloud and fog computing, International Journal of Distributed Sensor Networks, 13(11), pp.1-16

[39] Y. Sun, F. Lin, H. Xu (2018), Multi-objective optimization of resource scheduling in Fog computing using an improved NSGA-II, Wireless Personal Communication,102, pp. 1369–1385.

[40] Y. Yang, S.Zhao, W.Zhang, Y.Chen, X.Luo, J.Wang (2018), DEBTS: delay energy balanced task scheduling in homogeneous fog networks , IEEE Journal of Internet of Things, 5, pp 2094-2106.