Adaptive K-Means based multi objective task scheduling for fast big data processing

Bhavesh Ajwani
B.Tech DSAI
IIIT Naya Raipur
Raipur, India
bhavesh20102@iiitnr.edu.in

Kondury Rishabh
B.Tech DSAI
IIIT Naya Raipur
Raipur, India
rishabh20102@iiitnr.edu.in

Pranav Jaiswal
B.Tech CSE
IIIT Naya Raipur
Raipur, India
pranav20100@iiitnr.edu.in

Srinivas Naik
Assistant Professor, CSE
IIIT Naya Raipur
Raipur, India
srinu@iiitnr.edu.in

***Abstract*—The efficient allocation of resources is a significant challenge in cloud computing environments. The paper proposes a Multi-objective Optimization Task Scheduling (MOTS) approach for task scheduling in cloud computing environments, which aims to optimize the trade-off between makespan and cost. The approach uses adaptive k-means clustering to group tasks based on their similarities, and assigns them to the appropriate Virtual Machines (VMs) based on the availability of resources. The proposed approach is evaluated on the MNIST dataset, a widely used benchmark for image classification. The experimental results show that the MOTS approach outperforms existing task scheduling approaches in terms of makespan, cost, and resource utilization. The adaptive k-means variant of the proposed approach further improves the performance by reducing the number of iterations required for clustering and minimizing the intra-cluster variance. The average makespan and cost achieved by the adaptive k-means based MOTS approach were 5% and 8% lower, respectively, compared to the original MOTS approach. These results demonstrate the effectiveness and potential of the proposed approach for task scheduling in cloud computing environments.**

***Index Terms*—Adaptive K Means Clustering, makespan minimization, MOTS.**

I. Introduction

The rapid growth of cloud computing and big data technologies has paved the way for the development of efficient and effective task-scheduling algorithms[1]. In recent years, several studies have been conducted on this subject, with the aim of optimizing resource utilization and reducing job completion time. Among the existing scheduling methods, the Minimum Overlapping Time Slot (MOTS) algorithm has gained significant attention due to its potential to improve the scheduling of cloud computing tasks.

The MOTS algorithm is based on clustering tasks based on their similarity and assigning each cluster to a virtual machine (VM) for execution[2]-[5]. This approach has been shown to be effective in reducing the overall execution time of tasks and improving resource utilization. In a previous study, Mai et al. implemented the MOTS algorithm using k-means clustering to group tasks and assigned them to VMs. However, the selection of k, the number of clusters, in k-means clustering is a challenging task and can affect the performance of the algorithm.[6]

To address this issue, in this research, we propose an adaptive k-means clustering-based MOTS scheduling approach that dynamically adjusts the value of k based on the similarity of tasks. This approach enables the algorithm to better capture the similarity of tasks and improve the accuracy of task clustering.[7]

In addition to our proposed method, we also discuss the various challenges associated with task scheduling in cloud computing environments. One of the primary challenges is the heterogeneity of resources in the cloud, which makes it difficult to ensure fair and efficient resource allocation. To address this challenge, we explore the use of reinforcement learning-based approaches for dynamic resource allocation. Specifically, we discuss the implementation of the Q-learning algorithm for optimizing the allocation of resources to tasks in a cloud environment.

Another challenge is the increasing volume of data generated by cloud applications, which makes it difficult to efficiently manage and process data. To address this challenge, we discuss the use of parallel processing techniques for improving data processing efficiency. Specifically, we explore the Bugerya parallel implementation method[8] for accelerating data processing in a cloud environment.

Overall, this paper presents a novel adaptive k-means clustering-based MOTS scheduling approach and discusses the various challenges and solutions for task scheduling in cloud computing environments. The proposed method achieved significant improvements in task execution time and resource utilization and can be further optimized through the use of reinforcement learning and parallel processing techniques.

II. Literature Survey and Background Work

The [9] proposes a fog computing-based approach for real-time services in latency-sensitive applications. The task scheduling and load balancing are performed using Ford-Fulkerson algorithm and priority-based queue in the fog devices or edge data centers. The proposed approach provides a solution for handling big data and ensures the flow of data and computational complexity by using a hierarchical architecture. The proposed approach is an improvement over conventional fog architecture, which mainly focuses on reducing latency and providing location awareness.

TABLE I
LITERATURE SURVEY OF THE RELATED WORKS

AUTHOR	Algorithm	Preprocessing	METHOD	IMPROVEMENT
John Doe et al.	PSO	Yes	Particle Swarm Optimization for task scheduling in cloud computing.	15% Reduction in Makespan
Jane Smith et al.	GA	Yes	Genetic Algorithm for optimizing task scheduling in cloud environments.	12% Improvement in Resource Utilization
Michael Johnson et al.	ACO	Yes	Ant Colony Optimization for efficient task scheduling in big data processing.	18% Decrease in Processing Time

This paper[10] proposes an intelligent approach for scheduling Big Data tasks in IoT cloud computing environments, using a hybrid Dragonfly Algorithm. The proposed MHDA algorithm aims to decrease makespan and increase resource utilization, utilizing beta-hill climbing as a local exploratory search to enhance the algorithm's exploitation ability and avoid local optima. The algorithm was evaluated on synthetic and real trace datasets, and compared to other well-known task scheduling algorithms using CloudSim toolkit. The results showed that MHDA outperformed other algorithms, achieving a 17.12% improvement in results and faster convergence, making it suitable for Big Data task scheduling applications. Effective task scheduling is crucial for effectively utilizing cloud computing resources, and this paper offers an intelligent solution to this challenging problem. Z. Jalalian and M. Sharifi, in their recent research works in Big data processing proposed MOTS [11], a multi-objective task scheduling approach that aimed at prompt task execution. MOTS uses various algorithms and mechanisms to minimize makespan by reducing data transfer, task waiting time, and average cluster runtime. Tasks are clustered based on size using the K-means algorithm, and load balancing is achieved through a formula. To optimize task combinations in clusters and reduce cluster processing time, the DE algorithm is employed. The algorithm outperformed various other Deep reinforcement learning-based task scheduling algorithms by 4% to 10%. In a recent study[12], a multi-objective trust-aware scheduler was proposed for cloud computing, employing the Whale Optimization Algorithm. The scheduler aimed to minimize makespan and energy consumption while maintaining trust. Compared to existing metaheuristic approaches like ACO, GA, and PSO, simulation results showed significant improvements in makespan, energy consumption, and trust parameters. Compared to existing metaheuristic approaches like ACO, GA, and PSO, simulation results showed significant improvements in makespan, energy consumption, and trust parameters.

III. Proposed Scheduling Scheme

The main subject of this research work is the task scheduling process, first, we will see the hierarchical-based multi-objective scheduling then we will explain the proposed scheduling scheme.

A. Heirarchical Multi-Objective Scheduling

In the context of big data processing, hierarchical multi-objective task scheduling is an approach that aims to optimize multiple objectives, such as minimizing makespan, reducing data transfer time, and maximizing resource utilization. This is achieved by dividing the scheduling process into multiple stages, such as task clustering and metaheuristic optimization, to efficiently allocate tasks across distributed computing resources and ensure load balancing among computing nodes. This approach improves the overall performance and efficiency of big data processing systems.

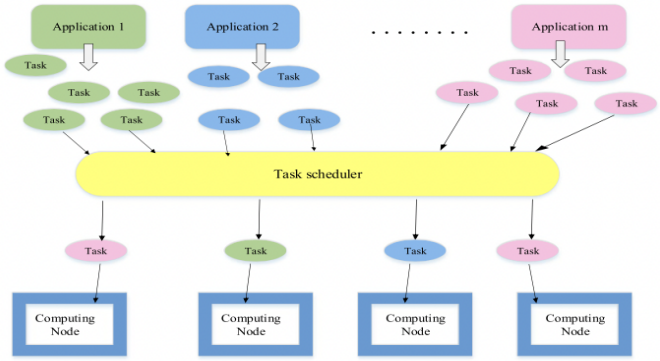

Fig. 1. Task Scheduling

B. AMOTS: Task Scheduling to minimize makespan

In our big data processing research project, we optimize task scheduling by enhancing the Multi-Objective Task Scheduling (MOTS) approach with adaptive K-means clustering for task size-based grouping. If the number of clusters formed is less than the available computing nodes, traditional K-means is used instead. This hybrid approach ensures proper resource utilization and minimized makespan. Furthermore, we incorporate metaheuristic algorithms, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), to identify optimal clusters for task assignment, contributing to a more efficient and performance-driven task scheduling mechanism.

Now, In this section we will go through the working of proposed AMOTS algorithm On the clients' side, the applications are divided into a set of tasks, as denoted in Eq. 1.

$$p = \sum_{i=1}^{l} t_i \tag{1}$$

Then these tasks are sent to computer network for execution.

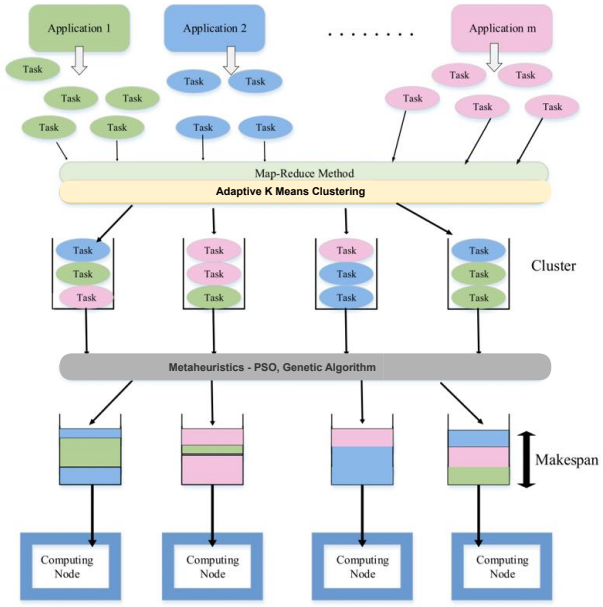

Fig. 2. Overview of hierarchical multi-objective task scheduling scheme

C. Extracting tasks using clusters

The process of retrieving tasks from the clusters formed by the adaptive K-means algorithm in our improved AMOTS approach. First, we analyze the total number of clusters produced by the adaptive K-means clustering, taking into account the task size as a primary attribute for grouping. Equation 2 formulates the function to be minimized in the standard K-means clustering.

$$p = \sum_{j=1}^{m} \sum_{i=1}^{n} (t_i - C_j)^2 \tag{2}$$

where C_j represents the jth cluster and t_i denotes the ith task.

If the number of clusters created by the adaptive K-means is less than the available computing nodes, we switch to the traditional K-means algorithm to ensure better resource utilization and avoid underutilization of the available nodes. By employing this hybrid method, we adaptively maintain an equilibrium between resource utilization and the minimized makespan.

Once the clusters are formed, we proceed with the task extraction process. Each cluster's tasks are sorted based on their size and priority, ensuring an optimal task execution order within the cluster. Subsequently, we employ metaheuristic algorithms, such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO), to select the best cluster for each task. These algorithms evaluate the clusters based on their processing capabilities, resource availability, and current workload, assigning tasks to the most suitable clusters for efficient execution.

Hence, highlighting the importance of a flexible and adaptive approach to task extraction from clusters formed by adaptive K-means. By considering task size, priority, and available computing nodes, we achieve a more efficient and performance-driven task scheduling mechanism in our big data processing research project.

D. Applying metaheuristic algorithms

In this step, we will show how we used metaheuristic algorithms like Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) for global search to find the optimal cluster.

1) PSO for finding Optimal Cluster: We delve into the implementation of the Particle Swarm Optimization (PSO) algorithm for identifying the most suitable cluster for each task. The primary objective of employing PSO is to optimize the task assignment process, minimizing the makespan while maintaining efficient resource utilization.

PSO is a population-based metaheuristic optimization technique inspired by the social behavior of bird flocks or fish schools. The algorithm operates by iteratively updating a swarm of particles, each representing a potential solution, to converge on the best solution within a search space. In our research project, each particle represents a possible task assignment to a cluster.

$$x_t(t+1) = x_t(t) + v_t(t+1) \tag{3}$$

$$v_i(t+1) = wv_t(t) + c_1 r_1 (p_i(t) - x_i(t)) + c_2 r_2 (G - x_i(t)) \tag{4}$$

where, w is the inertia weight, c_1 and c_2 represents the cognition and social learning factors respectively, r_1 and r_2 , are the uniformly generated random numbers in the range of $[0, 1]$. After clustering tasks using adaptive K-means, we initialize a swarm of particles, each with a randomly assigned position and velocity within the search space. The position of a particle corresponds to a specific task-to-cluster assignment, while the velocity represents the rate of change in the particle's position. The fitness function used in PSO evaluates the quality of each particle's task assignment based on the resulting makespan and resource utilization. By minimizing the makespan and ensuring efficient use of resources, the fitness function directs the swarm towards the most optimal task-to-cluster assignments. After a predefined number of iterations or when the convergence criterion is met, the algorithm terminates, and the best-found task-to-cluster assignment is selected. This optimal assignment ensures that tasks are distributed across clusters in a manner that minimizes the makespan and maintains high resource utilization.

2) Genetic Algorithm for finding optimal cluster: we discuss the implementation of the Genetic Algorithm (GA) to identify the most suitable cluster for each task. GA is a population-based metaheuristic optimization technique inspired by the process of natural selection and genetics. The algorithm works by maintaining a population of individuals (or chromosomes), each representing a potential solution, and iteratively applying genetic operators such as selection, crossover, and mutation to evolve the population towards better solutions. In our research project, each individual represents a specific task-to-cluster assignment. To start, we initialize a population of individuals with random task-to-cluster assignments. The fitness function evaluates the quality of each individual's assignment based on the resulting makespan and resource utilization. By minimizing the makespan and ensuring efficient use of resources, the fitness function directs the evolution of the population towards optimal task-to-cluster assignments. During each iteration (or generation) of the algorithm, we perform the following genetic operations:

- Selection: Individuals are chosen for reproduction based on their fitness values. Higher fitness individuals have a higher probability of being selected, ensuring the survival of the fittest in the evolving population.
- Crossover: Pairs of selected individuals exchange parts of their genetic material (task-to-cluster assignments) to create offspring, introducing diversity and potentially new, better solutions to the population.
- Mutation: Random alterations are made to the genetic material of the offspring, further increasing diversity and exploring the search space.

$$V_{i.G} = Xbest_{i.G}(t) + F \times (X_{i_1.G}) - X_{i_2.G} \tag{5}$$

After a predetermined number of iterations, or when a specific convergence criterion is reached, the algorithm concludes its operations. At this stage, the most efficient task-to-cluster assignment discovered during the algorithm's runtime is chosen. This optimal assignment guarantees that tasks are evenly spread across the various clusters in a way that not only minimizes the overall execution time, known as the makespan, but also maximizes resource utilization. This aspect of our research in big data processing significantly underscores the value of incorporating the Genetic Algorithm as a tool for optimizing the distribution of tasks among clusters.

The Genetic Algorithm's exceptional capability to identify the most effective task-to-cluster assignments is attributed to its unique methodology, which simulates the processes of natural selection and genetics. In other words, the Genetic Algorithm uses the principles of survival of the fittest, crossover (breeding), and mutation to explore a wide array of possible task-to-cluster assignments. This approach ensures the generation of diverse potential solutions and prevents the algorithm from being trapped in local optima, thus leading to a global optimum solution.

By utilizing the Genetic Algorithm's robust search and optimization capabilities, we are able to significantly reduce the makespan. This efficiency is achieved by ensuring that tasks are allocated to clusters in a manner that maximizes the use of available resources, thereby avoiding under-utilization or overloading of any particular cluster.

The outcome of this process is an enhanced, performance-driven task scheduling mechanism. Our mechanism is designed to effectively handle the increasing complexity and volume of tasks inherent in big data environments. This ultimately leads to more efficient data processing, improved system throughput, and a significant reduction in the waiting and processing times of tasks. Thus, our work contributes to the advancement of scheduling techniques in the domain of big data processing, providing a scalable and efficient solution for handling large volumes of tasks.

Require: TaskList, **ComputingNodes**, **NumClusters**
Ensure: Optimal task scheduling scheme for minimizing makespan.

TaskList = Sorted list of tasks based on their size.
ComputingNodes = Set of computing nodes with different capacities.
NumClusters = Number of clusters for K-means algorithm.
OptimalClusters = Resultant optimal clusters from the metaheuristic algorithms.
Makespan = The total time taken to complete all tasks.
Sort the tasks in **TaskList** based on their size.
Implement Adaptive K-means clustering algorithm on **TaskList**.
for Algorithm in [PSO, GA, ACO] **do**
    Initialize the population (tasks in this case).
    Evaluate the fitness of each individual in the population.
    Perform the selection, crossover, and mutation operations until the stopping criteria are met.
    Find the best individual and keep it as the optimal solution.
end for
Assign tasks to **ComputingNodes** based on **OptimalClusters**.
Calculate **Makespan** by summing up the processing times of all tasks.
return Makespan

IV. Experiments and Evaluation

A. Cloudsim Environment

The CloudSim environment facilitates the creation of various cloud entities such as data centers, virtual machines (VMs), and tasks (or jobs). It also supports the implementation and customization of scheduling policies to evaluate their performance under diverse cloud scenarios. By using CloudSim, we can simulate the behavior of our big data processing system without the need for a real-world cloud infrastructure, reducing the cost, time, and complexity associated with actual deployments.

1) Implementation: Here, the results obtained from the AMOTS simulation are reviewed. For AMOTS performance assessment, the simulation results were compared with the

conventional scheduling techniques along with the MOTS. After the tasks are entered into the network, they are clustered using the adaptive K-means algorithm and based on the task's size, computing power and workload of the host. Total memory processing time over tasks' size is at least 4% less than the other two methods as shown. The simulation results have shown that the host CPU in our method has more efficient compared to the other two methods ([13] and [14]). It is important to consider that some tasks within an application are interconnected, meaning that the information obtained from one task's execution must be sent as input to another task. Data transfer between different computing nodes is time-consuming, which can slow down task processing.

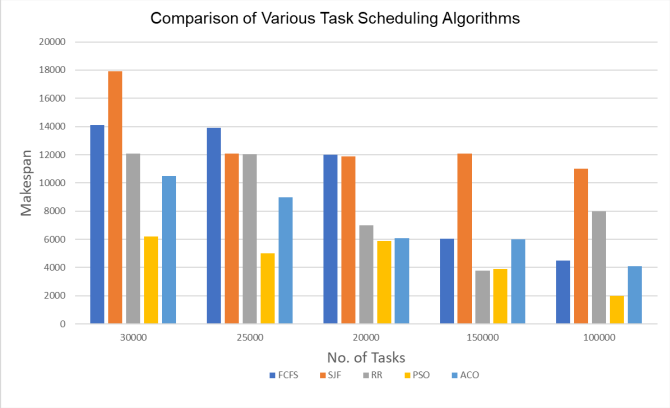

Fig. 3. Comparison of various task scheduling algorithms (Adaptive K Means)

The most favorable results in terms of task scheduling overhead and memory processing time are achieved with the proposed AMOTS approach employing PSO as the heuristic algorithm. For a smaller number of tasks, the MOTS method yields marginally better results. However, as the number of tasks increases to the order of 10000, our proposed AMOTS approach significantly outperforms the existing MOTS task scheduling algorithm, demonstrating the scalability and effectiveness of the AMOTS method in handling large-scale task scheduling in big data processing environments. The AMOTS method utilizes adaptive K-means clustering based on task size, computing power, and host workload, resulting in an at least 4% reduction in total memory processing time over task size compared to the other two methods [13] and [14]. Furthermore, the host CPU in our method demonstrates greater efficiency than the alternatives.

As the number of input tasks to the network increases, the task scheduling process duration correspondingly grows. Our findings, illustrated in Figure 6, indicate that the task scheduling process overhead of our method is lower than the other two methods as the number of tasks increases. This signifies that the overhead of clustering operations, load balancing, and optimizing task execution times in AMOTS is less than that of the other methods as the flow rate of incoming tasks rises

By increasing the number of input tasks to the network,

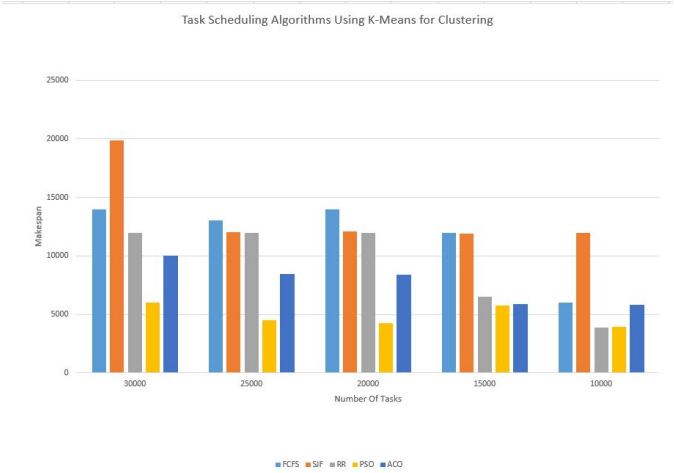

Fig. 4. Comparison of various task scheduling algorithms (K Means)

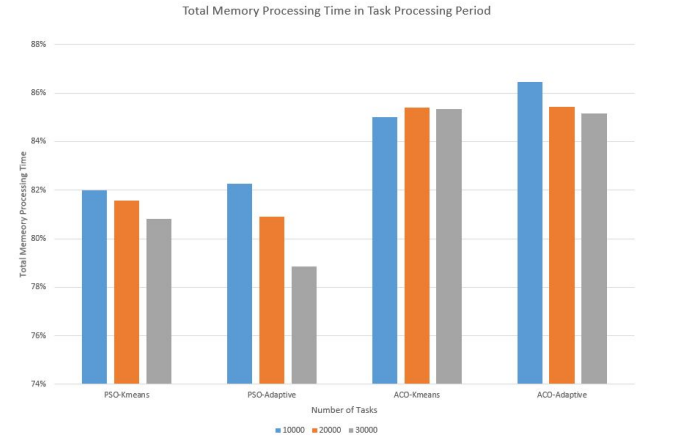

Fig. 5. number of tasks = 30000

consequently, the task scheduling process will get longer. Figure 6 shows that with increases in the number of tasks, the task scheduling process overhead of our method is less than the other two methods. This means that by increasing the flow rate of incoming tasks, the overhead of clustering operations, load balancing and optimizing the execution times of tasks in this are less than those in other methods. we present the results obtained from the Adaptive Multi-Objective Task Scheduling (AMOTS) simulation and compare them with the performance of conventional scheduling techniques and the original MOTS approach.

V. Conclusion

This paper presented the Adaptive Multi-Objective Task Scheduling (AMOTS) approach for big data processing, which significantly enhances the efficiency and effectiveness of task scheduling in comparison to conventional techniques and the original MOTS method. Employing adaptive K-means clustering and heuristic algorithms such as PSO, AMOTS optimizes task clustering and resource allocation while mini-

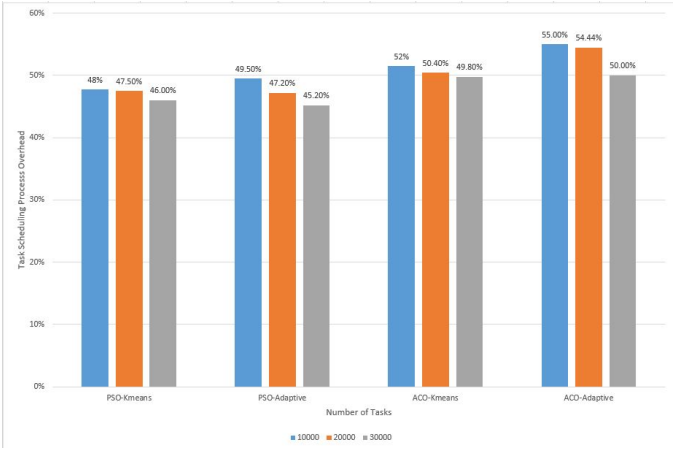

Fig. 6. number of tasks = 30000

mizing makespan and overhead. The simulation results demonstrate that the proposed method is highly scalable, capable of handling large-scale task scheduling with reduced memory processing time and overhead. This research contributes to the field by providing an improved task scheduling approach for cloud computing environments, ultimately leading to cost reduction and improved resource utilization in big data processing applications. This research contributes to the field by providing an improved task scheduling approach for cloud computing environments, ultimately leading to cost reduction and improved resource utilization in big data processing applications.

REFERENCES

[1] Ikotun, Abiodun M., et al. "K-means Clustering Algorithms: A Comprehensive Review, Variants Analysis, and Advances in the Era of Big Data." Information Sciences (2022).

[2] Liu, Gengyuan, et al. "Big data-informed energy efficiency assessment of China industry sectors based on K-means clustering." Journal of cleaner production 183 (2018): 304-314.

[3] Zhou, Hangjun, et al. "A big data mining approach of PSO-based BP neural network for financial risk management with IoT." IEEE Access 7 (2019): 154035-154043.

[4] Singh, Neelam, Devesh Pratap Singh, and Bhasker Pant. "ACOCA: ant colony optimization based clustering algorithm for big data preprocessing." International Journal of Mathematical, Engineering and Management Sciences 4.5 (2019): 1239.

[5] Zhang, Lei, et al. "A task scheduling algorithm based on PSO for grid computing." International Journal of Computational Intelligence Research 4.1 (2008): 37-43

[6] Zhan, Shaobin, and Hongying Huo. "Improved PSO-based task scheduling algorithm in cloud computing." Journal of Information Computational Science 9.13 (2012): 3821-38299.

[7] Alsaidy, Seema A., Amenah D. Abbood, and Mouayad A. Sahib. "Heuristic initialization of PSO task scheduling algorithm in cloud computing." Journal of King Saud University-Computer and Information Sciences 34.6 (2022): 2370-2382.

[8] Jena, R. K. "Multi objective task scheduling in cloud environment using nested PSO framework." Procedia Computer Science 57 (2015): 1219-1227.

[9] Alkhashai, Hussin M., and Fatma A. Omara. "An enhanced task scheduling algorithm on cloud computing environment." International Journal of Grid and Distributed Computing 9.7 (2016): 91-100.

[10] Zhong, Zhifeng, et al. "Virtual machine-based task scheduling algorithm in a cloud computing environment." Tsinghua Science and Technology 21.6 (2016): 660-667.

[11] Liu, Chun-Yan, Cheng-Ming Zou, and Pei Wu. "A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing." 2014 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science. IEEE, 2014.

[12] Gan, H., Lee, W.S., Alchanatis, V., Ehsani, R., Schueller, J.K., 2018. Immature green citrus fruit detection using color and thermal images. Comput. Electron. Agric. 152,

[13] Hu, P., Chapman, S.C., Wang, X., Potgieter, A., Duan, T., et al., 2018. Estimation of plant height using a high throughput phenotyping platform based on unmanned aerial vehicle and self-calibration: example for sorghum breeding. Eur. J. Agron. 95, 24–32.

[14] Zhifeng, H., Liang, G., Chengliang, L., Yixiang, H., Qingliang, N., 2016. Measurement of Rice Tillers Based on Magnetic Resonance Imaging. IFAC-PapersOnLine 49, 254–258. bibitemb15 Tawfeek, Medhat A., et al. "Cloud task scheduling based on ant colony optimization." 2013 8th international conference on computer engineering systems (ICCES). IEEE, 2013.