

## **A Review and Assessment of Nature-Inspired and Machine Learning–Based Task Scheduling Algorithms in Cloud Computing**

**Bharat Chhabra**

*Research Scholar, CSE Department, MAIT, Maharaja Agrasen University, Baddi, Himachal Pradesh 174103 (India)*

**Dr. Pankaj Nanglia**

*Professor, Maharaja Agrasen University, Baddi, Himachal Pradesh 174103 (India)*

**Dr. Neha Kishore**

*Assoc. Prof., Maharaja Agrasen University, Baddi, Himachal Pradesh 174103 (India)*

**Abstract:** The evolution of task scheduling algorithms in cloud computing environments has been a crucial aspect for efficient resource utilization and improved performance. The paper analyses the various task scheduling algorithms that have been proposed over the years, including the First-Come-First-Served (FCFS) algorithm, Shortest Job First (SJF) algorithm, and Round Robin (RR) algorithm etc. After exploring the available literature on more advanced algorithms such as the Load Balancing Algorithm, Priority Scheduling Algorithm, and Latest heuristics based hybrid Scheduling Algorithm till date to the best of our knowledge, this keynote paper proposes a novel classification of all contemporary and latest task scheduling algorithms. Finally, the paper concludes by highlighting the detailed merits and demerits of each latest approach viz. soft-computing techniques, Machine Learning techniques and other nature-inspired techniques.

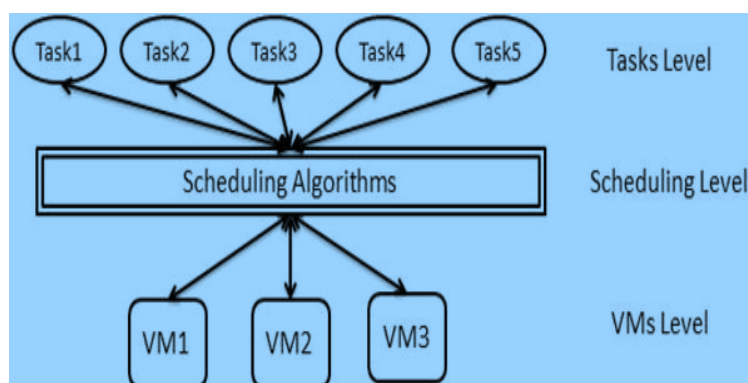
**Keywords:** Cloud Computing, algorithms, machine learning techniques etc.

### **1. INTRODUCTION**

The cloud computing environment consists of a collection of connected computers or nodes that have an access to the pool of shared resources (e.g., applications, networks, servers, storage, and services), information, and software over the internet based on their requirement anywhere, anytime on “pay-per-utilize” basis [1]. In recent years, due to exaltation in communication technology, incendiary use of internet, the cloud computing technology is showing unexceptional rise. Cloud computing technology helps in fast arrangement of inter-connected geographically disseminate data centres for providing high quality services thus empowering the clients without to be concerned about server setups and configurations to execute their applications. Cloud computing paradigm offers various services that are usually categorised in such as software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS). Cloud computing environment may be deployed as private, public, hybrid and community cloud.

In the runtime environment of cloud computing, scheduling is a crucial aspect [2]. Scheduling refers to the approach through which user tasks allocated to the virtual instances of physical machines to uplift the overall efficiency and quality of service. From the data analysis of last decade, there is continuously increase in the number of requests for cloud services thus also

increasing the workload on cloud environment. An inefficient scheduling method results in performance degradation because of cloud resources being underutilized or overutilized or remains idle. Scheduling algorithms always strive to attain the primary goal of higher efficiency by allocating the task on servers to attain other goals of balanced load among all virtual instances, higher utilization of resources, lower execution time of user task thus increases the system throughput, saves energy, decreases cost and many more.



**Fig. 1: Typical Task Scheduling Process**

In simple terms, (Fig. 1) represents the typical arrangement of VMs and incoming tasks involves allocating resources and tasks to processing elements to optimize the use of the cloud infrastructure [3]. This can include allocating VMs to different users, prioritizing tasks based on their importance or urgency, and managing the workloads of different VMs to ensure that they are running at optimal levels.

## 2. LITERATURE REVIEW

**A. Particle Swarm Optimization (PSO)** is a descendant of meta-heuristic optimization techniques that are applied to task-resource mapping problem in CCE. Many researchers have applied this technique to find the solution.

1. In [11], authors have applied PSO to schedule tasks in cloud computing settings; the paper introduces a hybrid technique which merges the merits of PSO algorithm and Cuckoo Search algorithm. This hybrid algorithm has reduced task execution time and allocates resources more efficiently. As per the results after simulating this hybrid algorithm using Cloudsim simulator, the proposed algorithm performed efficiently over others in terms of both precision and efficacy. Unfortunately, the approach is only useful for small-scale systems, and using it with large-scale systems might necessitate more modifications.

2. Authors in [12] used a load-balancing method and their research suggests an improved algorithm based on PSO to figure out the task-resource mapping in CCE. With a view to enhance resource usage and cut down on job completion time, the suggested approach seeks to optimize task allocation and scheduling. The simulation results demonstrate the improvement in completion time, resource consumption, load balancing and the proposed algorithm performs more effectively than the initial PSO algorithm and other task scheduling techniques currently in use. According to the research authors, the suggested algorithm can considerably raise the efficacy and efficiency of cloud computing systems.

3. The article in [13] proposed an adaptive algorithm based on PSO method and incorporates an adaptive load balancing approach. This algorithm aims to improve task scheduling efficiency by optimizing the allocation of computing resources, while also adapting to changing workload demands. Results proved that suggested algorithm was able to perform well than existing algorithms, with lower task completion times and more efficient resource utilization. The study concludes that the PSO-based approach with adaptive load balancing has potential for improving cloud computing performance in real-world applications.

4. The paper [14] proposed another adaptive PSO algorithm called AdPSO for task scheduling in CCE. AdPSO uses a dynamic neighbourhood topology and a mutation operator to enhance the search process, which results in improved makespan, energy consumption, and utilization of resources compared to existing methods. The algorithm adapts to the changing workload and resource availability by adjusting its parameters accordingly. Simulation results show that AdPSO outperforms other state-of-the-art scheduling algorithms in with regard to efficiency and scalability. The proposed approach has the potential to be applied to other optimization problems in cloud computing.

5. The paper [15] proposed an improved particle swarm optimization algorithm to do task-resource mapping in grid environment. The proposed algorithm utilizes a new encoding method for job scheduling and incorporates a dynamic job arrival mechanism. The PSO technique was used to optimize the makespan by finding the optimal allocation of resources to jobs. The simulation results have proven the higher performance of proposed algorithm as makespan was lowered; resource utilization went higher and lowered the job completion time also.

These papers illustrate the different ways PSO algorithm has been used for task-resource mapping in CCE. However, this is worth noting that the specific problem being solved, the environment, and the metrics used to evaluate the performance of the algorithm vary from one study to another.

**B. Ant Colony Optimization (ACO)** is a meta-heuristic algorithm which is widely applied for the problems of task-resource mapping in CCE. Researchers have used this technique at various instances and have proved it to perform better.

1. In order to schedule tasks in cloud computing environment, the article [16] suggests slave ants-based Ant Colony Optimization (ACO). The algorithm uplifts the performance of task scheduling by dividing all tasks into multiple subsets and assigning each subset to a group of slave ants. The slave ants use local search methods to find the optimal solution for their assigned subset, and the global pheromone trail is updated based on the best solutions found by the slave ants. Using CloudSim tools, the scheduling method was simulated and the findings indicated that it performed better than alternative scheduling methods.

2. The study [17] proposed an Ant Colony Optimization (ACO) model aiming the improvement in QoS while doing task-resource mapping with randomly generated tasks and cloud resources. The model aimed to reduce makespan, total completion time, and maximize the QoS of tasks and the results showed their success in terms of makespan and QoS. The suggested model's weakness, however, is that it cannot handle dynamic task arrivals and departures, which may affect its performance in real-world scenarios.

3. [18] Proposed an algorithm that was based on Ant Colony Optimization technique for task scheduling in heterogeneous multiprocessing environments. The proposed algorithm uses the MATLAB simulation tool with the aims to minimize the makespan by intelligently allocating tasks to processors with different computing capacities. To assess the proposed algorithm's performance and compare it to other algorithms, the authors ran a number of experiments. After completing numerous trials, the study concluded with better makespan and execution time than existing ones. The presented algorithms has scaling limits, though, as it gets less effective as the number of jobs and processors rises.

4. [19] An adaptive scheduling approach for cloud tasks using ant colony optimization (ACO) is proposed in the paper. The authors used the CloudSim toolkit to simulate the proposed approach with the aim to optimize the scheduling of tasks to improve the overall performance of cloud computing systems and evaluate its performance. The analysis revealed better performance of suggested algorithm than conventional scheduling methods when compared with timeliness, energy usage, and resource usage. Yet the study has some flaws, namely its limited scalability and disregard for communication delays. Overall, the study provides insights into the potential benefits of using ACO for cloud task scheduling.

5. Research in [20] proposed an efficient and scalable task scheduling algorithm that was based on Ant Colony Optimization technique and authors implemented the proposed algorithm in Java using the CloudSim simulation toolkit for green cloud computing environments and evaluated its performance on a real-world cloud dataset. For scheduling jobs, the suggested algorithm considers the energy usage of cloud data centres, task completion times, and work transfer costs. According to the findings, the suggested algorithm performs better than traditional scheduling algorithms as it proved to lower energy usage, lower job completion time, and lesser migration expense. However, this study had some limitations too, such as the absence of a comparison with commercial cloud providers, and the need for further testing on larger datasets.

These are just a few examples of recent research papers that have used ACO for task-resource mapping in CCE. It's worth noting that the specific implementation of ACO, the parameters used, and the results obtained vary depending on the specific research paper and use case.

**C. Honey Bee algorithm (HBA)** is a descendant of nature-inspired optimization technique resembling the foraging conduct of honey bees. HBA technique is applied to optimize certain objectives, such as minimal completion time or maximizing resource utilization. Here are a few examples of recent research papers that have used HBA for task scheduling:

1. Paper [21] presented a Honey Bee inspired algorithm for optimal distribution of tasks in CCE and it applied "swarm intelligence," where individual agents (similar to bees in a hive) work together to achieve a common goal. The algorithm's performance is compared to existing ones as compared on makespan and resource utilisation using MATLAB simulation tool. The use of a simulated environment and the necessity for future testing in a real-world cloud computing environment are inherent challenges of the study, too.

2. Research in paper [22] proposed a load and fault aware HBA, which aims to optimize task allocation and average usage of resources. Proposed algorithm is inspired by honey bee foraging behavior and is designed to consider both the current load on the cloud and potential faults that may occur. Simulations are used to assess the algorithm and when compares with other two task scheduling algorithms it excels in areas of task completion time and resource utilization. But, have some flaws, such as using a modelling tool and the lacking of a real-world implementation.

3. The article [23] proposed a novel algorithm for balanced distribution of tasks based on the behaviour of honey bees where worker bees balance their workload based on the amount of nectar available in different locations. The algorithm uses virtual bees to represent virtual machines in the cloud and directs them to the least loaded physical machine. The results from the simulation tool, CloudSim, indicate that the suggested algorithm works better than conventional load balancing methods in terms of accelerating response times and optimizing resource usage. The study's scalability and algorithmic generalisation, however, are both constrained.

4. A load balancing technique proposed in [24] based on the Min-Min scheduling algorithm combined with honey bee algorithm. This approach aims to optimize the task-resource mapping in CCE that improved system performance and reduced response time. The experiments conducted using the CloudSim simulator has shown that the proposed technique outperformed traditional load balancing methods, such as Round Robin and Ant Colony Optimization. The method was not scalable, and the load balancing mechanism must be often reconfigured as a result.

5. A new honey bee algorithm to schedule tasks in a cloud environment by considering both load and fault-awareness is proposed in [25]. In comparison to other scheduling algorithms, the proposed technique optimizes them in terms of makespan, energy usage, and fault-tolerance. The algorithm's applicability in larger, more complicated cloud settings is unclear, though, as the study only takes into account a single data centre scenario.

In summary, these papers proposed the usage of honey bee algorithm for task scheduling in cloud computing. These algorithms were able to minimize the completion time of tasks,

maximizing the utilization of resources and ensuring efficient usage of resources. But it's worth noting that the performance of the algorithm can depend on the specific use case and the design of the algorithm.

**D. Grey Wolf Optimization (GWO) algorithm** is a meta-heuristic algorithm that has been used in various research papers for scheduling the tasks in CCE. Some recent research papers that have used GWO have proved to perform better.

1. The research in [26] proposed a new approach for that used Pareto-based Grey Wolf Optimizer (PGWO). The PGWO algorithm is a combination of the Grey Wolf Optimizer (GWO) and Pareto optimization to address the limitations of existing methods. The algorithm was evaluated on a real-world dataset using MATLAB tool and contrasted with other approaches. Analysis indicated that PGWO performed efficiently when compared on various parameters viz. timeliness, cost, and resource usage. The study, however, only evaluated the algorithm on a single dataset and neglected to take other crucial aspects like security and reliability into account.

2. A new strategy using the Grey Wolf Optimizer (GWO) was presented in [27] with the aims to minimize the makespan of the tasks and the consumption of energy. Performance of proposed task scheduler was assessed using a simulation environment and the evaluation has shown the effectiveness of proposed GWO-based scheduler as it reduced the makespan and lowered the energy consumption when compared with three other well-known task scheduling algorithms. The use of simulations and a lack of implementation in the real world are two challenges of the proposed approach.

3. [28] Proposed an improved Grey Wolf Optimization (GWO) based algorithm with objectives of lowering makespan and total cost of executing tasks. Simulation results using MATLAB R2015a proved that proposed algorithm performed better than many existing algorithms, including standard GWO algorithm, Particle Swarm Optimization (PSO), and Genetic Algorithm (GA) and outperformed. Although the study's consideration of static work scheduling scenarios is a restriction, it still excels other studies in terms of makespan and cost.

4. Another study in [29] proposed Grey Wolf Optimization (GWO) algorithm for load-balancing in CCE which considers reliability as a critical performance factor. The algorithm was evaluated using MATLAB tool for simulation experiments and compared with other well-known load balancing algorithms. The findings demonstrated that in terms of dependability, throughput, and response time, the suggested approach performs better than existing algorithms. The study however had certain drawbacks, including the use of a particular set of parameters and limited scalability.

5. A modified Grey Wolf Optimizer algorithm was proposed in [30] that optimized task-resource mapping solution with total completion time than conventional task scheduling algorithms. The modified grey wolf optimizer used the MATLAB software to improve the search for the optimal solution. The work does, however, had certain weaknesses, such as the use of just one performance parameter and the failure to take into account the dynamic nature of cloud computing systems.

It's worth noting here that these researches are just symbolic examples and there are many more papers that have used GWO algorithm for task scheduling in cloud computing environments.

**E. ABC (Artificial Bee Colony) algorithm** is a meta-heuristic technique based on impersonating the conduct of bees and it is applied to many domains for solving optimization problems, including cloud computing.

1. Paper [31] presented a study on how artificial bee colony (ABC) algorithm is used to optimize the allocation of virtual machines in CCE. Authors suggested a new approach to schedule virtual machines in the cloud that reduced both the cost and duration of job execution. The study concluded that the proposed method is effective in reducing makespan and execution cost of

tasks. However, study's scope was constrained, and it left out crucial details like resource utilization and security.

2. The bee colony technique was used in [32] to balance workloads in CCE. It simulated the behaviour of bees in finding the best food sources, and is used to optimize the allocation of tasks to cloud resources. The authors evaluated the proposed approach using the CloudSim simulator and comparison of results with other algorithms have shown that the bee colony algorithm outperformed while balancing the load. However, there remained certain drawbacks to the study, such as the failure to take into account communication cost and network delay.

3. The suggested approach using ABC in [33] worked well to effectively distribute jobs across the available VMs with objectives of lowered makespan and overall energy usage. The scheduling strategy is evaluated using simulation experiments, and the analysis shows that it performs better than other scheduling algorithms in terms of effectiveness and power usage. The study, however, does not involve a real-world implementation and is instead restricted to simulation experiments.

4. Firefly algorithm was used to propose a novel algorithm in [34] and it was verified through formal verification and simulation analysis using MATLAB simulator. The algorithm outperformed existing algorithms in terms of makespan and energy consumption. However, the limitation of the study was that it did not consider network latency, which can affect the performance of the algorithm.

These papers have shown that the application of Artificial Bee Colony technique is well suited to the problem of task-scheduling in CCE. However, certain drawbacks are yet required to be attended in each of the mentioned algorithms.

**F. The Bat Algorithm (BA)** is another meta-heuristic technique motivated from the conduct of bats and has been used in various research papers for figuring out optimal task-resource mapping solution in CCE.

1. With an objective to optimise energy usage, the study in [35] suggested the use of meta-heuristic optimization technique i.e. Bat algorithm which is motivated by the echolocation behaviour of bats. Proposed strategy sought to reduce energy usage and increase resource utilisation in cloud computing environments. According to the experimental findings, the suggested approach performed better in terms of efficiency and energy usage than other current algorithms. The study was constrained using a single objective function and the assumption of homogenous resources.

2. Authors in [36] came up with a novel approach for load balancing in cloud computing environments using Bat Algorithm (BA). The strategy sought to evenly distribute workload among virtual machines to reduce response time and maximise resource usage. According to the findings, the suggested algorithm achieved the objectives of lower response time and higher resource usage than already-used methods like Round Robin and ant colony optimization. The proposed method, however, was only applicable to certain workload characteristics and did not take the heterogeneity of the cloud environment into account.

3. [37] Proposed a hybrid algorithm that combined the strengths of PSO and Bat Algorithm. Using the simulation tool MATLAB, the Ranging and Tuning PSO with BA (RT-PSO-BA) algorithm is tested on different datasets and compared with other scheduling techniques. The proposed hybrid approach resulted optimal makespan, resource utilization, and energy consumption. The use of simulated datasets and the absence of real-world cloud applications were two of the study's drawbacks, though.

4. The authors in [38] proposed a bat algorithm-based energy-aware workflow scheduling and optimization approach for cloud computing. The proposed approach optimizes energy consumption by scheduling workflows to the appropriate resources based on their energy efficiency. The approach was evaluated using CloudSim simulator, and the results indicated that

it can effectively optimize the energy consumption of cloud-based workflows. The study's weaknesses, meanwhile, were its limited sample size and use of homogeneous cloud resources. Yet, the proposed methodology offers a potential path that is energy-efficient.

5. With an objective of higher efficiency, the study in [39] offered an innovative method for multi-objective task scheduling in a cloud computing environment that combines the bat algorithm with a local search methodology. CloudSim tool was used for evaluating the experiments and compared against several state-of-the-art algorithms. The outcomes revealed that the hybridized bat algorithm achieved better makespan, lowered the carbon emission and workload was also evenly distributed. The algorithm was constrained by the complexity of the objective functions and its capacity to scale.

These studies clearly indicate that the usages of algorithms that are derived from Bat algorithms are of great importance while solving the problem of task-resource mapping in CCE. It may be used independently or in conjunction with other existing nature-inspired algorithms.

**G. The Machine Learning** methods for task scheduling have been in use since the last decade for solving many problems in various domains including the problem of task-scheduling in cloud computing:

1. Research in [40] suggested a reinforcement-learning based algorithm that used multiple directed acyclic graphs (DAGs) for workflow scheduling in cloud computing. The algorithm took into consideration the dependencies between tasks and assigns them to resources in an efficient manner. Based on the results, it could be seen that the suggested method surpasses conventional scheduling algorithms on various parameters including makespan, resource utilization and completion time. Proposed strategy also had several shortcomings, though, such as the lack of tests with large-scale DAGs and comparisons with more sophisticated scheduling techniques.

2. The OCSA algorithm presented in [41] is a task scheduling method designed for cloud computing environments. It aimed to optimize the distribution of computing resources and maximize performance. The algorithm used a priority queue to select tasks based on their priority and available resources, and assigned them to appropriate virtual machines. According to the study, OCSA was more efficient overall and accomplish better results than well-known scheduling algorithms on resource consumption and job completion time parameters. The approach was not scalable and does not handle task dependencies.

3. In the research work of [42], a novel scheduling algorithm called QL-HEFT that made use of machine learning approaches was proposed to optimize task allocation in cloud computing environments. QL-HEFT took into account not only the computational requirements of tasks but also the dynamic characteristics of cloud resources, and targeted for optimal makespan and improve resource usage. Algorithm was evaluated by the authors using a number of simulated scenarios, and its results were contrasted with those of other scheduling techniques and it excelled in both makespan and resource utilization. Yet, the study's scalability and applicability in the actual world were both hindered.

4. [43] Proposed a machine learning-based technique specifically developed for uplifting healthcare related facilities-cum-services that may be distributed on cloud computing platforms. The model aimed to predict patient outcomes based on their medical history and to optimize resource allocation for better healthcare delivery. The limitations of the proposed model included the need for high-quality data, privacy concerns, and the difficulty of deploying the model on a large scale. The article concluded that the model has the potential to enhance healthcare services, but further research was required to overcome the limitations and assess the effectiveness of the model.

5. The paper [44] presented a scheduling scheme for cloud computing environments using deep Q-learning. The proposed scheme was designed to optimize the allocation of resources by

considering the dynamic nature of cloud computing environments. The results displayed lesser time was spent on job completion, and algorithms made best use of available resources. Nevertheless, it only succeeded in homogenous computing systems and was unable to deal with unforeseen occurrences that might happen in cloud computing environments.

**H. Genetic Algorithm [GA]** is like of evergreen help to improve the performance of algorithms designed to achieve single or multiple objectives in parallel. Similar is the case of task-resource mapping algorithms in CCE.

1. Authors presented a GA based algorithm to find out task-resource mapping in [45]. Study aimed to optimize resource utilization and minimize the makespan. When compared with round-robin and first-come first-serve scheduling algorithms, the suggested method performed much better in terms of efficiency and reliability. But the study was limited to a single objective optimization.

2. An algorithm for genetically based job task-resource mapping solution in cloud computing systems was presented in the paper [46]. The algorithm aimed to optimize task scheduling by considering the computational resource availability and task characteristics. The algorithm was evaluated using CloudSim simulator and compared with three other scheduling algorithms and it performed more effectively when compared on important criteria of makespan and usage of resources. However, presented evaluation was restricted to a single cloud environment and did not consider other factors such as energy consumption and security.

3. The research in [47] introduced a new approach for task-resource mapping problem using a GA family algorithm that was both context and load-aware. The algorithm was developed to optimize resource-utilization in the cloud and to minimize the impact on other tasks being performed on the same platform. Using the simulation programme MATLAB, this novel method exceeded the other methods in terms of efficiency, energy use, and resource use. The lack of real-world testing and the small number of nodes used in the simulation are the major drawbacks.

4. The authors in [48] proposed a GA inspired solution for Fog computing environments. The proposed approach i.e. Genetic Algorithm-Improved Resource-Aware Cost-efficient Scheduler (GA-IRACE), aimed to optimize many parameters such as makespan, higher utilization of resources and even distribution of load to available VMs. The suggested algorithm was simulated using MATLAB, and its results were compared to alternative scheduling techniques with reference to makespan, resource utilisation and cost. The resulted data demonstrated that GA-IRACE worked better. However, the study had drawbacks because it only took into account one goal at a time and neglected network latency.

5. Research in [49] proposed another algorithm that was based on a combination of genetic algorithm and ant colony optimization to figure out optimal task-resource mapping solution. The proposed algorithm was simulated using MATLAB with objectives of lower execution time, higher usage of resources. The simulation results have proven that algorithm presented has surmounted the existing task scheduling algorithms in regards of completion time and resource utilization. However, this study's shortcomings include the absence of real-world application and the requirement for more study in a variety of circumstances.

### **3. CLASSIFICATION OF TASK SCHEDULING ALGORITHMS**

In Cloud computing, there are many different algorithms and techniques for task scheduling, including static, dynamic, heuristic and meta-heuristic algorithms. These algorithms can be designed to optimize different factors such as resource utilization, response time, or energy consumption. Some of the well-known task scheduling algorithms in cloud computing are First-Fit, Best-Fit, Worst-Fit, and Round-Robin [4].

Selection of a specific task scheduling algorithm is influenced by task's needs, the type of cloud environment, and the available resources.

**A. Depending upon the time of decision** for actual mapping of task with the resource, existing algorithms for task scheduling may be grouped into two categories:

1. Static scheduling algorithms: Static scheduling algorithms are a type of algorithms that does not change itself according to requirement or environment. It requires the complete information regarding the user tasks like number of tasks, length of tasks, deadlines of tasks etc and resources (storage requirement, power consumption, processing requirement etc) required to complete the tasks. Static scheduling algorithms are not the good choice where there is a continuous change or frequent fluctuations in the workload but this happens in cloud environment. So here dynamic algorithms play a better role. Examples are: round robin scheduling, “min-min [5]” algorithms, “max-min [6]” algorithms, FIFO algorithm, SJF algorithm etc.

2. Dynamic scheduling algorithms: These algorithms do not need prior information about the workload and resources required to complete the tasks. These algorithms continuously monitor the system so that load can be balanced properly on different machine and resources can be efficiently utilized between the various tasks. For the faster execution of tasks dynamic algorithms shifts the load from the over loaded machine to under loaded machine and for this these algorithms continuously monitor the system and change itself according to requirement. Examples are: “particle swarm optimization (PSO) [7]”, “ant colony optimization (ACO) [8]”, Genetic algorithms [9], “Honey-bee algorithms” and many more. All these come under soft computing techniques.

**B. Depending upon the technique** used for finding the task and resource mapping, another such classification may be heuristic algorithms and meta-heuristic algorithms.

1. Heuristic techniques are commonly used in cloud computing environments to schedule tasks due to their ability to quickly find near-optimal solutions to complex problems. Here are a few popular heuristic techniques used:

i) First-Fit Decreasing (FFD): This technique sorts the tasks in descending order of their resource requirements and then assigns them to the first available resources that can accommodate them.

ii) Best-Fit Decreasing (BFD): Similar to FFD, BFD sorts the tasks in descending order of their resource requirements, but it assigns each task to the resource that can accommodate it with the least amount of waste.

iii) Round Robin (RR): This technique assigns tasks to resources in a cyclic manner. Tasks are assigned to the next available resource in a circular list of resources.

iv) Randomized Priority: This technique assigns a random priority value to each task, schedules them according to such assigned priority values.

Each technique mentioned above has its own strengths and weaknesses and the best technique to use will depend on the specific requirements of the task scheduling problem at hand.

2. Meta heuristic techniques are high-level problem-independent techniques that enable the algorithm to find approximate resolutions to complex tasks based on optimization, including task scheduling in cloud computing environments [8]. Here are a few popular meta-heuristic techniques that are available for scheduling the tasks in CCE:

i) Simulated Annealing (SA): This technique is inspired by the annealing process of metals and uses a random search algorithm to find near-optimal solutions to the task scheduling problem. The algorithm gradually reduces the temperature over time, allowing the search to converge to the optimal solution.

ii) Particle Swarm Optimization (PSO): This method is stimulated by the behavior of flocks of birds that uses a population of particles to explore the solution space. Here particle signifies a possible answer to the task scheduling problem, and they move in solution space based on positions of their neighbors.

iii) Differential Evolution (DE): This technique uses the principles of natural evolution to generate new possible resolutions to the resource allocation problem. The algorithm generates new possible solutions after merging the characteristics from existing solutions.

iv) Artificial Bee Colony (ABC): This technique is inspired by the foraging behavior of honeybees and uses a population of artificial bees to explore the answer. Here, bee signifies a possible answer to task scheduling mapping, and the bees move in the solution space based on the positions of their neighbors.

v) Bat Algorithm (BA): This technique is inspired by the echolocation behavior of bats and uses a population of bats to explore the answer. Every bat signifies a possible answer to task scheduling mapping, and the bats move in the solution space based on the positions of their neighbours.

It's worth noting that the choice of algorithm relies on specific needs of incoming tasks and system being used. Additionally, the scheduling algorithms can be combined and hybridized to achieve better performance and results.

**C. Depending upon the reference model** that is taken as an analogy to decide on actual task and resource mapping, this class of task scheduling algorithm belongs to a common broad category known as soft computing. This is a division of AI which leads to the development of task scheduling solutions that can solve problems that are hard to solve using traditional methods. Task scheduling in cloud computing environment is a crucial issue that can benefit from the use of soft computing techniques. There are a variety of Soft computing-based approaches for the solution to the problem of task scheduling in cloud computing environment such as genetic algorithms, fuzzy logic and neural networks. These techniques have proven to enhance the efficiency of algorithms [10].

1. Fuzzy logic, for example, can be used to elevate the performance of task-resource mapping solution. In this approach, fuzzy sets and fuzzy rules are used to represent the uncertainty and imprecision of the task requirements and resource availability. A fuzzy logic system can then make decisions based on this uncertain information, resulting in more efficient resource allocation.

2. Genetic algorithms, on the other hand, use principles of natural selection and genetics to evolve a population of solutions over time. By simulating the process of natural selection, genetic algorithms are able to optimize the task-resource mapping in CCE.

3. Artificial neural networks (ANNs) also used to schedule tasks in cloud computing environment. In this approach, ANNs are used to model the relationship between the task requirements and the resource availability. By learning from historical data, the ANN can predict the resource requirements of new tasks, which is used to schedule the tasks efficiently.

4. Ant Colony Optimization (ACO): ACO algorithm is inspired from the actions of ant colonies. It is used to uplift the efficiency of task-resource mapping solution due to its ability to find near-optimal solutions in a reasonable amount of time.

In conclusion, soft computing-based algorithms are particularly well-suited for solving problems that are hard to solve using traditional methods.

**D. Depending upon machine learning techniques**, this latest class of algorithms and methods enable computers to learn from data, rather than being explicitly programmed. These techniques are suitable to figure out approximate solutions to complex optimization problems, including task-resource mapping in CCE. Machine learning-based scheduling is increasingly being put to practice to optimize task scheduling in cloud environments. Researchers have proposed various ML-based scheduling algorithms, such as neural network-based scheduling, deep reinforcement learning-based scheduling, and evolutionary algorithms-based scheduling. These algorithms can

learn from historical data and adapt to changing resource requirements and workloads. Some popular machine learning methods used for task scheduling in cloud computing environments:

1. Reinforcement learning: This type of machine learning technique is used to train an agent to make decisions in an environment with the goal of maximizing some form of reward. In cloud computing, this technique can be used to train an agent to make scheduling decisions that optimize resource utilization and minimize costs.
2. Neural networks: Neural networks are a type of machine learning technique that are modelled after the structure and function of the human brain. They can be used to learn complex relationships between inputs and outputs, and are often used to predict future events or classify data. In cloud computing, neural networks can be used to predict workloads and make scheduling decisions accordingly.
3. Genetic Algorithm: This technique is inspired by the principles of natural evolution and is generally used within a machine learning technique for feature selection or hyper-parameter tuning etc. The algorithm generates fresh possible solutions by merging the characteristics of existing solutions. They can be used to find the optimal scheduling in cloud computing environment.
4. Support Vector Machine (SVM): This is a supervised linear learning model that uses classification algorithms and applies to regression problems as well. In the context of cloud computing, it can be used for task scheduling by training a model to predict the best resources for a given task, based on historical data.
5. Random Forest: Random forest is a collaborative learning model that combines the output from two decision trees into one to reach out a new decision. It can be used to make scheduling decisions by analyzing data on historical resource utilization and task performance.

These are a few examples of machine learning techniques that are applicable for task-resource mapping problem in CCE. Each technique has its own strengths and weaknesses, and the best technique to use will depend on the specific requirements of the task scheduling problem at hand. Also, Machine learning techniques are more computationally expensive than heuristic and meta-heuristics but often provide better solutions.

#### 4. CONCLUSION

Algorithms	Merits	Demerits
<b>Static task scheduling algorithms:</b>	<ol style="list-style-type: none"> <li>1. The schedules are predetermined and therefore predictable, which can be useful for certain types of workloads.</li> <li>2. The scheduling process is relatively simple and can be completed quickly.</li> <li>3. There is no need to constantly monitor the system and make adjustments to the schedule.</li> </ol>	<ol style="list-style-type: none"> <li>1. The schedules are not adaptable to changes in the workload or resource availability.</li> <li>2. The schedules may not be optimal for all types of workloads or resource configurations.</li> <li>3. Static scheduling algorithms may not be able to handle dynamic workloads or changing resource availability.</li> </ol>
<b>Dynamic task scheduling algorithms:</b>	<ol style="list-style-type: none"> <li>1. The schedules can be adjusted in real-time to respond to changes in the workload or resource availability.</li> <li>2. Dynamic scheduling algorithms can handle dynamic workloads and changing resource availability.</li> <li>3. They can provide better performance and resource utilization compared to static scheduling algorithms.</li> </ol>	<ol style="list-style-type: none"> <li>1. The scheduling process can be more complex and time-consuming.</li> <li>2. There may be a need for more monitoring and adjustments to the schedule.</li> <li>3. Dynamic scheduling algorithms may be less predictable than static scheduling algorithms.</li> </ol>
<b>Heuristic algorithms:</b>	<ol style="list-style-type: none"> <li>1. Heuristics can find a near-optimal solution in a reasonable amount of time.</li> <li>2. Heuristics can be applied to a wide</li> </ol>	<ol style="list-style-type: none"> <li>1. Heuristics may not find the optimal solution, but only a near-optimal one.</li> <li>2. They may be sensitive to the initial</li> </ol>

	<p>range of scheduling problems.</p> <p>3. Heuristics can be easily implemented and integrated into existing systems.</p>	<p>conditions and parameters.</p> <p>3. They may not be able to handle large and complex scheduling problems.</p>
<b>Meta-heuristic algorithms:</b>	<p>1. Meta-heuristics can find the optimal solution or a near-optimal solution in a reasonable amount of time.</p> <p>2. Meta-heuristics can be applied to a wide range of scheduling problems.</p> <p>3. Meta-heuristics can be easily implemented and integrated into existing systems.</p>	<p>1. Meta-heuristics may require a lot of computational resources.</p> <p>2. They may be less predictable than heuristics.</p> <p>3. They may not be able to handle large and complex scheduling problems.</p>
<b>Machine Learning Algorithms:</b>	<p>1. They are robust to the initial conditions and parameters.</p> <p>2. Improved Resource Utilization: Optimize the use of available resources to improve efficiency and reduce cost.</p> <p>3. Dynamic Adaptation: Able to adapt to changes in workload and resource availability on the fly.</p> <p>4. Improved Performance: Efficient scheduling can lead to improved overall system performance.</p>	<p>1. Complexity: Can be complex to implement and maintain.</p> <p>2. Overhead: Additional computational overhead required for scheduling can impact system performance.</p> <p>3. Lack of Guaranteed Performance: Can't guarantee a certain level of performance for all tasks due to dynamic adaptation.</p> <p>4. Resource Contention: Can lead to resource contention and reduced overall system performance.</p>
<b>Genetic Algorithm (GA):</b>	<p>1. GA can handle a large number of variables and constraints</p> <p>2. GA can find a near-optimal solution</p>	<p>1. GA may get stuck in a local optimum</p> <p>2. GA can be slow when dealing with large datasets.</p>
<b>Particle Swarm Optimization (PSO):</b>	<p>1. PSO is easy to implement and understand</p> <p>2. PSO can handle a large number of variables and constraints</p>	<p>1. PSO may get stuck in a local optimum</p> <p>2. PSO can be sensitive to the choice of parameters</p>
<b>Ant Colony Optimization (ACO):</b>	<p>1. ACO can handle a large number of variables and constraints</p> <p>2. ACO can find a near-optimal solution</p>	<p>1. ACO may get stuck in a local optimum</p> <p>2. ACO can be sensitive to the choice of parameters</p>
<b>Bat Algorithm (BA):</b>	<p>1. BA can handle a large number of variables and constraints</p> <p>2. BA can find a near-optimal solution</p>	<p>1. BA may get stuck in a local optimum</p> <p>2. BA can be sensitive to the choice of parameters</p>
<b>Bee Algorithm (BEA):</b>	<p>1. BEA can handle a large number of variables and constraints</p> <p>2. BEA can find a near-optimal solution</p>	<p>1. BEA may get stuck in a local optimum</p> <p>2. BEA can be sensitive to the choice of parameters</p>
<b>Firefly Algorithm (FA):</b>	<p>1. FA can handle a large number of variables and constraints</p> <p>2. FA can find a near-optimal solution</p>	<p>1. FA may get stuck in a local optimum</p> <p>2. FA can be sensitive to the choice of parameters</p>
<b>Reinforcement Learning:</b>	<p>1. Can learn from experience and adapt to changing conditions.</p>	<p>1. Can be computationally expensive, and can struggle to converge on optimal policies.</p>
<b>Neural Networks:</b>	<p>1. Can model complex relationships between inputs and outputs, and can handle large amounts of data.</p>	<p>1. Can be computationally expensive, and can struggle with overfitting to training data.</p>

<b>Decision Trees:</b>	1. Simple to understand and interpret, and can handle both numerical and categorical data.	1. Can be prone to overfitting, and can struggle with large datasets or complex relationships between inputs and outputs.
<b>Support Vector Machines (SVMs):</b>	1. Can handle large datasets and non-linear relationships between inputs and outputs.	1. Can be computationally expensive, and can struggle with large amounts of noisy data.
<b>Bayesian Methods:</b>	1. Can handle uncertain data and incorporate prior knowledge.	1. Can be computationally expensive, and can struggle with high-dimensional data.

The exhaustive coverage of these algorithms discloses various merits and demerits of each algorithm. This insight leads to the dark sides of the algorithms and also the application areas of each one of these. From the above analysis, it can be concluded that despite being dynamic, Machine learning algorithms do impose little overhead on the machines and also lack in guaranteed upliftment in results. Genetic algorithms are suitable to handle large number of variables and constraints but suffer from the problem of local optimum. Heuristics based algorithms has a wider applicable area but may not always find an optimum solution. PSO and ACO based algorithms are sensitive to choice of parameters. Neural networks motivated algorithms can handle large amount of data but are computationally expensive. Hence, it can be concluded that none of the class of algorithm is suitable to all types of task scheduling problems as the choice of a particular algorithm is influenced by many associated factors.

## 5. REFERENCES

1. R. Buyya, C.S. Yeo, S. Venugopal, "Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities", "10th IEEE International Conference on High Performance Computing and Communications, HPCC 2008, Dalian, China", Sept. 2008.
2. P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *Recomm. Natl. Inst. Stand. Technol.*, no. Special Publication 800-145, p. 7, Sep. 2011, doi: 10.6028.NIST.SP.800-145.
3. Fang Liu, Jin Tong, Jian Mao, Robert Bohn, John Messina, Lee Badger and Dawn Leaf, "NIST Cloud Computing Reference Architecture", "NIST Special Publication 500-292", pp. 1-28, September 2011.
4. D. Villegas, A. Antoniou, S.M. Sadjadi and A. Iosup, "An analysis of provisioning and allocation policies for infrastructure-as-a-service clouds", 12th IEEE/ACM International Symposium on cluster, cloud and grid computing, 2012, pp. 612-619.
5. L. Agostinho, G. Feliciano, L. Olivi, E. Cardozo, and E. Guimaraes, "A Bio-inspired Approach to Provisioning of Virtual Resources in Federated Clouds," in 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Sydney, Australia, pp. 598–604, Dec. 2011. doi: 10.1109/DASC.2011.109
6. O. M.Elzeki, M. Z. Reshad, and M. A. Elsoud, "Improved Max-Min Algorithm in Cloud Computing," *Int. J. Comput. Appl.*, vol. 50, no. 12, pp. 22–27, Jul. 2012, doi: 10.5120/7823-1009
7. K. Dubey and S. C. Sharma, "A novel multi-objective CR-PSO task scheduling algorithm with deadline constraint in cloud computing," *Sustain. Comput. Inform. Syst.*, vol. 32, p. 100605, Dec. 2021, doi: 10.1016/j.suscom.2021.100605
8. H. Singh, S. Tyagi, P. Kumar, S. S. Gill, and R. Buyya, "Metaheuristics for scheduling of heterogeneous tasks in cloud computing environments: Analysis, performance evaluation, and future directions," *Simul. Model. Pract. Theory*, vol. 111, p. 102353, Sep. 2021, doi: 10.1016/j.simpat.2021.102353

9. M. Mezmez et al., "A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems," *J. Parallel Distrib. Comput.*, vol. 71, no. 11, pp. 1497–1508, Nov. 2011, doi: 10.1016/j.jpdc.2011.04.007
10. E. H. Houssein, A. G. Gad, Y. M. Wazery, and P. N. Suganthan, "Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends," *Swarm Evol. Comput.*, vol. 62, p. 100841, Apr. 2021, doi: 10.1016/j.swevo.2021.100841
11. Al-maamari, Ali, and Fatma A. Omara. "Task scheduling using hybrid algorithm in cloud computing environments." *Journal of Computer Engineering (IOSR-JCE)* 17.3 (2015): 96-106
12. Ebadifard, Fatemeh, and SeyedMortezaBabamir. "A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment." *Concurrency and Computation: Practice and Experience* 30.12 (2018): e4368
13. Ahmad, MdOqail, and Rafiqul Zaman Khan. "Pso-based task scheduling algorithm using adaptive load balancing approach for cloud computing environment." *International Journal of Scientific & Technology Research* 8.11 (2019)
14. Nabi, Said, et al. "AdPSO: adaptive PSO-based task scheduling approach for cloud computing." *Sensors* 22.3 (2022): 920
15. Patel, Surendra K., and Anil Kumar Sharma. "Improved PSO based job scheduling algorithm for resource management in grid computing." *International Journal of Advanced Technology and Engineering Exploration* 6.54 (2019): 152-161
16. Moon, YoungJu, et al. "A slave ants based ant colony optimization algorithm for task scheduling in cloud computing environments." *Human-centric Computing and Information Sciences* 7.1 (2017): 1-10
17. Sharma, N., & Garg, P. (2022). Ant colony based optimization model for QoS-Based task scheduling in cloud computing environment. *Measurement: Sensors*, Volume 24, 100531
18. Elcock, Jeffrey, and Nekiesha Edward. "An efficient ACO-based algorithm for task scheduling in heterogeneous multiprocessing environments." *Array* (2023): 100280
19. Mishra, Sambit Kumar, BibhudattaSahoo, and P. Satya Manikyam. "Adaptive scheduling of cloud tasks using ant colony optimization." In *Proceedings of the 3rd International Conference on Communication and Information Processing*, pp. 202-208. 2017
20. Ari, Ado Adamou Abba, IrepranDamakoa, ChafiqTitouna, Nabila Labraoui, and AbdelhakGueroui. "Efficient and scalable aco-based task scheduling for green cloud computing environment." In *2017 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 66-71. IEEE, 2017
21. LD, DhineshBabu, and P. Venkata Krishna. "Honey bee behavior inspired load balancing of tasks in cloud computing environments." *Applied soft computing* 13.5 (2013): 2292-2303
22. Gupta, Punit, and Satya Prakash Ghrera. "Load and fault aware honey bee scheduling algorithm for cloud infrastructure." In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014: Volume 2*, pp. 135-143. Springer International Publishing, 2015
23. Hashem, Walaa, HebaNashaat, and RawyaRizk. "Honey bee based load balancing in cloud computing." *KSII Transactions on Internet and Information Systems (TIIS)* 11.12 (2017): 5694-5711

24. Thapliyal, Nitin, and PritiDimri. "Load Balancing in Cloud Computing Based on Honey Bee Foraging Behavior and Load Balance Min-Min Scheduling Algorithm." *International Journal of Electrical and Electronics Research (IJEER)* 10.1 (2022): 1-6
25. Gupta, Punit, and Satya Prakash Ghrera. "Load and fault aware honey bee scheduling algorithm for cloud infrastructure." In *Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications (FICTA) 2014: Volume 2*, pp. 135-143. Springer International Publishing, 2015
26. Khalili, Azade, and SeyedMortezaBabamir. "Optimal scheduling workflows in cloud computing environment using Pareto-based Grey Wolf Optimizer." *Concurrency and Computation: Practice and Experience* 29.11 (2017): e4044
27. Bacanin, Nebojsa, TimeaBezdan, Eva Tuba, Ivana Strumberger, Milan Tuba, and MiodragZivkovic. "Task scheduling in cloud computing environment by grey wolf optimizer." In *2019 27th telecommunications forum (TELFOR)*, pp. 1-4. IEEE, 2019.
28. Natesan, Gobalakrishnan, and ArunChokkalingam. "An improved grey wolf optimization algorithm based task scheduling in cloud computing environment." *Int. Arab J. Inf. Technol.* 17.1 (2020): 73-81
29. Sefati, SeyedSalar, MaryamsadatMousavinasab, and Roya ZarehFarkhady. "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation." *The Journal of Supercomputing* 78.1 (2022): 18-42
30. Alzaqebah, Abdullah, Rizik Al-Sayyed, and Raja Masadeh. "Task scheduling based on modified grey wolf optimizer in cloud computing environment." In *2019 2nd International Conference on new Trends in Computing Sciences (ICTCS)*, pp. 1-6. IEEE, 2019
31. Kruekaew, B., and W. Kimpan. "Virtual machine scheduling management on cloud computing using artificial bee colony." In *Proceedings of the International MultiConference of engineers and computer scientists*, vol. 1, pp. 12-14. 2014
32. Babu, KR Remesh, Amaya Anna Joy, and Philip Samuel. "Load balancing of tasks in cloud computing environment based on bee colony algorithm." In *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)*, pp. 89-93. IEEE, 2015
33. Navimipour, NimaJafari. "Task scheduling in the cloud environments based on an artificial bee colony algorithm." In *International conference on image processing*, pp. 38-44. 2015
34. Ebadifard, Fatemeh, Saeed Doostali, and SeyedMortezaBabamir. "A firefly-based task scheduling algorithm for the cloud computing environment: Formal verification and simulation analyses." In *2018 9th International Symposium on Telecommunications (IST)*, pp. 664-669. IEEE, 2018
35. Sharma, Shabnam, Ashish Kr Luhach, and S. A. Sinha. "An optimal load balancing technique for cloud computing environment using bat algorithm." *Indian J SciTechnol* 9.28 (2016): 1-4
36. Valarmathi, R., and T. Sheela. "Ranging and tuning based particle swarm optimization with bat algorithm for task scheduling in cloud computing." *Cluster Computing* 22 (2019): 11975-11988
37. Gu, Yi, and ChanduBudati. "Energy-aware workflow scheduling and optimization in clouds using bat algorithm." *Future Generation Computer Systems* 113 (2020): 106-112
38. Bezdan, Timea, et al. "Multi-objective task scheduling in cloud computing environment by hybridized bat algorithm." *Journal of Intelligent & Fuzzy Systems* 42.1 (2022): 411-423

39. Cui, Delong, WendeKe, Zhiping Peng, and JinglongZuo. "Multiple DAGs workflow scheduling algorithm based on reinforcement learning in cloud computing." In Computational Intelligence and Intelligent Systems: 7th International Symposium, ISICA 2015, Guangzhou, China, November 21-22, 2015, Revised Selected Papers 7, pp. 305-311. Springer Singapore, 2016
40. Krishnadoss, Pradeep, and Prem Jacob. "OCSA: task scheduling algorithm in cloud computing environment." International Journal of Intelligent Engineering and Systems 11.3 (2018): 271-279
41. Tong, Zhao, et al. "QL-HEFT: a novel machine learning scheduling scheme base on cloud computing environment." Neural Computing and Applications 32 (2020): 5553-5570
42. Abdelaziz, Ahmed, et al. "A machine learning model for improving healthcare services on cloud computing environment." Measurement 119 (2018): 117-128
43. Tong, Zhao, et al. "A scheduling scheme in the cloud computing environment using deep Q-learning." Information Sciences 512 (2020): 1170-1191
44. Kumar, Marish, and Amit Doegar. "RELIABLE AND EFFICIENT TASK SCHEDULING ALGORITHM BASED ON GENETIC ALGORITHM IN CLOUD COMPUTING ENVIRONMENT." International Journal of Computing and Corporate Research ISSN (Online): 2249-054X Volume 5 Issue 6 November 2015, International Manuscript ID : 2249054XV5I6112015-02
45. Hamad, Safwat A., and Fatma A. Omara. "Genetic-based task scheduling algorithm in cloud computing environment." Int. J. Adv. Comput. Sci. Appl 7.4 (2016): 550-556
46. Kaur, Kamaljit, Navdeep Kaur, and Kuljit Kaur. "A novel context and load-aware family genetic algorithm based task scheduling in cloud computing." In Data Engineering and Intelligent Computing: Proceedings of IC3T 2016, pp. 521-531. Springer Singapore, 2018
47. Arshed, Jawad Usman, et al. "GA-IRACE: Genetic Algorithm-Based Improved Resource Aware Cost-Efficient Scheduler for Cloud Fog Computing Environment." Wireless Communications and Mobile Computing 2022 (2022)
48. Liu, Chun-Yan, Cheng-Ming Zou, and Pei Wu. "A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing." 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, pp. 68-72. IEEE, 2014