[1]Sedigheh Yeganeh

[2]Abdulbaghi Ghaderzadeh

[3]Sadoon Azizi

Providing A Scheduling Program for Work and Machine Allocation in the Cloud Environment by Combining Meta-Engineering and Graph Theory Hybrid Algorithms

JES Journal of Electrical Systems

Abstract: - Cloud computing is known as a dynamic service provider using highly scalable and virtualized resources on the Internet. Many industries have started offering cloud services on a "pay-as-you-go" basis. Advances have led to the concept of marketplace exchanges, which enable trade between cloud providers and consumers. Lightweight and Platform are presented as independent frameworks called "marketplace environments" that allow consumers and providers to trade computing resources according to their needs. Resource by optimizing the amount of processing resources needed by cloud infrastructure customers. Therefore, our goal is to find the optimal number of virtual machines and how to properly arrange cloud services on them by observing service quality criteria such as response time and service delivery error. In this article, we are going to design and implement a new scheduling system with evolutionary algorithms combined with genetic algorithm, so that we can improve the scheduling problem in dependent tasks to a great extent. Finally, the combined algorithms proposed by the GA_SA algorithm have shown impressive efficiency. Is. In addition to being successful in improving the completion time of tasks in the cloud system, this algorithm has also been shown to be better than other evolutionary algorithms in the time flow of tasks. Another feature of this algorithm is to have a suitable speed to find the global optimum.

Keywords: Parallel genetic algorithm, refrigeration algorithm, optimal assignment of work and machine, cloud, dynamic decision tree

1- Introduction

With the rapid growth of information and communication technology, a large percentage of organizations and companies have moved towards the cloud computing paradigm, which provides cost-effective computing for resource-intensive applications. Cloud computing provides the possibility of using cloud resources as a useful tool for allocating resources through cloud computing analysis. Cloud computing is a suitable way to provide computing resources. With cloud computing, software environments and services are based on shared relationships. Instead of purchasing a license, users pay a monthly fee and the software and platforms are managed by the providers and continuously updated to maximize their performance and security.

Cloud processing is a computer model that tries to facilitate the access of users based on the type of demand they have from information and computing resources. This model tries to meet the needs of users with the least need for human resources, reducing costs and increasing the speed of information access [1] One of the techniques required to increase the flexibility and scalability of cloud data centers. Tozan is a burden. Load balancing is done with various goals of tolerance against failure, energy management, reducing response time and increasing service quality, and finally maintenance of servers. [2-3]

Currently, load balancing in cloud computing systems is considered a serious challenge because a distributed solution is always needed and it is not always possible to keep one or more servers idle and inactive just to fulfill some requests or to It is not economical. It is clear that due to the scale and complexity of these systems, it is impossible to assign centralized workers to specific servers, therefore, for the proper management of the service provider's resources, we need a load balance that is suggested to the service provider. With the advancement of information technology, there is a need to perform computing tasks everywhere and anytime. There is also a need for people to be able to do their heavy computing work without having expensive hardware and software, and cloud computing has been the latest technological answer to these issues[4-5].

[1] Department of computer engineering, Islamic Azad University, Sanandaj Branch, Sanandaj, Iran

[2]Department of computer engineering, Islamic Azad University, Sanandaj Branch, Sanandaj, Iran

[3]Department of Computer Engineering and IT, University of Kurdistan, Sanandaj, Iran

Due to the ever-increasing growth of requests and the joining of new customers to the world of computing, computing systems must also change and be more powerful and flexible than before. In such a case, users try to access a service based on their needs and regardless of where a service is contracted or how it is delivered. One of the best computing systems that tries to provide such services to users is cloud computing[6-7].

In the meantime, cloud computing has been presented as a model beyond a system that currently has the ability to respond to most requests and requirements. The flexible infrastructure of cloud computing and virtualization technology has provided new facilities to support business activities. But being in the competitive market is a very important issue for cloud service providers, and therefore they try to have secure and flexible data centers by using different management, security, computing and storage techniques.

The methods provided for load distribution are divided into two general categories:

A: Algorithms whose goal is to equalize the processing load between nodes.

B- Algorithms whose purpose is to prevent some nodes from being idle when they have queued requests in other groups.

which finally makes all the nodes finish their work at the same time. A good load balancing algorithm should be fast and not add heavy load to the nodes and at the same time have a correct view of the overall state of the system.

Load balancing technique based on genetic refrigeration algorithm can develop and formulate its own problems. It delivers two types of input data such as processor speed and the allocated load of the conversion and an output such as the balanced load that is needed in the system inference. The two parameters together are used to evaluate the balanced load on the data centers of the cloud computing environment. They are used through the genetic refrigeration algorithm.

The main goal of work scheduling is to shorten the time to complete the work and increase the operational efficiency and efficient and appropriate allocation of resources to the work. Due to the fact that the previous methods in cloud computing used less number of parameters, therefore, in this research, in order to improve the results, we use more criteria such as speed, reliability, cost in order to optimally allocate multiple queues of resources to tasks.

2-litterreview

Today, many studies have used genetic algorithms. For example, Rahmani et al. used genetic algorithms in the optimal design of an earthing system[8]. Zamani et al. used genetic algorithms in the design of noise reduction in medical X-ray images using wavelets and neural networks[9]. Vares et al. used genetic algorithms in integrated preventive maintenance planning and production planning for a single machine[10].

Sangeetha et al. (2022) [11] addressed the resource management framework using deep neural networks in multi-cloud environment, in operationalizing multi-cloud environments so that the cloud computing system can perform its tasks in the best way. And to return the answer in the shortest possible time and at high speed with cloud management, it should use an optimal algorithm. In this way, grid users pay the owners of those resources a financial fee for using the resources. This framework motivates the owners of the resources to share their resources. Different algorithms have been used for scheduling tasks in the cloud environment, such as fault-tolerant scheduling algorithm, genetic algorithm, ant colony algorithm, minimum algorithm, min-min algorithm, and heating algorithm. Unfortunately, the dynamism and heterogeneity of cloud resources cause the complexity of task scheduling. Most of the scheduling systems available in the cloud environment optimize the completion time of tasks separately.

Abu Aligah and Al-Kharabsheh (2022) [12] presented a modified multi-objective hybrid optimizer with genetic algorithm to solve the task scheduling problem in cloud computing. This paper proposes an efficient optimization method for job scheduling based on a hybrid multi-objective optimizer with a genetic algorithm called MVO-GA. The proposed MVO-GA is proposed to increase the performance of task transfer through cloud network based on the workload of cloud resources. It is necessary to provide sufficient migration decisions to reschedule the migration tasks based on the performance weight of the aggregated tasks in the cloud. The proposed method (MVO-GA) works on several characteristics of cloud resources: speed, capacity, job size, number of tasks, number

of virtual machines and throughput. The proposed method successfully optimizes the task scheduling of a large number of tasks (i.e., 1000-2000). The proposed MVO-GA had promising results in optimizing the transfer time of large cloud tasks, indicating its effectiveness. The proposed method has been evaluated based on the use of the cloud simulation environment using the MATLAB trustless system.

The authors (2012) [13] investigated maximizing the efficiency of the scheduling algorithm. In the article, the scheduling technique is used. By distinguishing it into a profit function and a loss function for a single task, it also uses the turnaround time tool effectively to increase productivity. An overall improvement in resource utilization and reduced processing cost is also demonstrated.

Maguluri et al. (2012) [14] considered a stochastic model based on load balancing and scheduling in cloud computing clusters, where tasks arrive according to a random process and request resources such as memory, CPU, and storage space. It takes join-theshortest-queue (JSQ) routing algorithms and two-choice routing with Max-Weight scheduling policy. The paper [16] proposed a market-based resource management policy based on the game theory model.

Kang et al. (2013) [15] proposed a new cloud resource management algorithm called CRAA/FA (Cloud Resource Allocation Through Fitness Auction Algorithm), which creates a market for cloud resources and allows resource agents and service brokers to He bargains in that market.

The authors [17] proposed a cloud management framework that allocates infrastructure resources to spot markets to best match customer demand in terms of supply and price to maximize provider revenue and customer satisfaction.

Dynamic sources were studied in [18]. It is based on efficient system optimization and provisioning of heterogeneous virtual machines. A random auction mechanism that efficiently allocates resources according to users' bids is presented. The objective of this mechanism is to minimize users' budget costs and VM delays, which the main concern is in IaaS cloud provisioning. In order to solve various challenges in workflow applications, [19] combined resource provisioning and scheduling algorithms such as Particle Swarm Optimization (PSO) to solve the overall workflow execution cost and deadline constraints.

Basahel and Yamin (2022) [20] have proposed a new genetic algorithm for efficient scheduling in the cloud environment. The cloud computing environment has enabled many applications and users in many fields. The cloud environment can be used for different task scheduling algorithms for efficient use of resources. The goals of task scheduling are to maximize resource utilization time, reduce task execution time, achieve high system throughput, and improve load balancing. Many optimization algorithms exist in the literature to efficiently schedule tasks independently to improve completion time, reduce execution cost, and certainly maximize resource utilization. The purpose of this paper is to present a genetic algorithm that can be used to design and implement an evolutionary system for efficient task scheduling. The proposed algorithm can create a management strategy and optimal use of resources in the Cloud environment. Such a system will be very useful to reduce the time of execution of work and the level of work.

Raman and Wahab (2022) [21] have calculated workflow scheduling using back-propagation neural network in cloud computing. To measure the efficiency of workflow scheduling, it is necessary to determine the construction time and execution cost. This paper proposes a priority-based back-propagation neural network (PBF-NN) hybrid scheduling algorithm for accurate measurement of construction time and execution cost. Backfill algorithm is used to schedule tasks to available resources. The percentage of migration using this algorithm is reduced compared to the First Come First Server algorithm. Then, Berger's model is used to measure the fairness of resource allocation. The system decides the reallocation of work based on fair value. The backpropagation neural network performs the virtual machine placement process with necessary training and testing. The proposed algorithm dynamically allocates tasks and reduces resource usage. We use an experimental study to demonstrate how the proposed system enables higher efficiencies in cost, manufacturing, and performance.

Paper [22] Efficient hybrid job scheduling optimization (EHJSO) uses cuckoo search optimization and gray wolf job optimization (GWO). Due to the forced parasitism of some cuckoo species (laying eggs in the nests of other species), an optimization approach for cuckoo search was developed. Gray Wolf Optimization (GWO) is a population-based artificial intelligence system inspired by gray wolf social structure and hunting strategies.

Table (1) comparison of timing algorithm

Function	Online auction framework [23]
Based on auction mechanisms to optimize system efficiency	Meta-heuristic optimization technique [24]
based on particle swarm optimization (pso) to minimize the overall cost of running the workflow	A2Sc combination algorithm [25]
Based on A2SC algorithm to reduce the cost of Vms services.	Failure-aware resource provisioning algorithm [26]
Based on a failure-aware provisioning algorithm to redirect user requests to appropriate cloud providers	supply based on demand, availability of resources, [27]
Based on ASFLA (Augmented Shuffled Frog Leaping Algorithm) technique to minimize running cost/time	Providing high bandwidth, low delay, low bit error rate and energy [28]
Based on polynomial energy-aware routing algorithm, integer linear programming (ILP) and heuristic to design energy-aware routes and allocate servers and switches to meet traffic requests.	Fault tolerance, fault detection and fault recovery [24]
Explored various techniques such as self-healing, self-diagnosis, preemptive migration, checkpoint, restart, replication, system node recovery, and job migration techniques.	Hybrid approach [25]
A combination of autonomous computing concept and RL (reinforcement learning) based agent to predict the future needs of cloud services	Online auction framework [26]

Table (2) comparison of types of scheduling algorithms

Disadvantages	Advantages	Time complexity	Purpose and characteristics	Algorithms
- Not taking into account the time of performing tasks	- Simplicity	O(m)	- Sending the task to the processor with the least load	Prohibited search algorithm
- Time to complete the above work	- Maximum use of resources	O(m)	Goal: keeping the nodes occupied as much as possible, or in other words, load balance between resources	Minimum execution time algorithm
- Failure to pay attention to the availability of resources and processors	- The load balance	O(m)	- Assignment of the task to a machine with the shortest execution time	Minimum Algorithm
- Load imbalance	-	O(mlogm)	- Send the task to the machine with the shortest completion time	completion time
- Assignment of tasks to a processor with inappropriate execution time	- Try to balance	$O(T^2 \times m)$	- Combination of OLB and MET methods	The best K-% algorithm

- Not choosing the correct value of k affects the work result	- Low task completion time	$O(T^2 \times m)$	Objective: To be given the best car for each task	Min-Min Algorithm
- Improper result if there are long tasks among short tasks	- The best completion time compared to the above algorithms	$O(T^2 \times m)$	- Considering a set of processors for allocation	Algorithm
	- Fair completion time	$O(m)$	- Allocation of work to a processor with the shortest completion time	Min-Max
	- Assign the machine/task pair selected in the first phase to a machine with less time	$O(T^2 \times m)$	- Has two phases of minimization of completion time	Algorithm
	- It compensates the damage caused by the implementation of long tasks		- Like the MCT method based on minimum completion time	Max-Min
- The possibility of hunger	- Completion time almost same as MCT and KPB	$O(n^2)$	Goal: Reduce task completion time by early allocation of small tasks to resources	Heating algorithm

3-modeling

Traditional algorithms, such as the round-robin algorithm, have problems that make them difficult to use. One of the most important problems was the lack of ability to balance the load and the lack of guarantee of finding the right solution for the problem. These problems occur due to the systematic and deterministic behavior of these types of algorithms. To solve these problems, innovative algorithms such as genetic algorithm and sa algorithm were introduced. Due to the population-based behavior, the genetic algorithm provided the possibility of finding suitable solutions and did not have the two problems of traditional algorithms, but this algorithm is not suitable for problems with a high degree of non-linearity, such as the scheduling problem. Because the possibility of the algorithm getting stuck in local optima is very high. On the other hand, the algorithm, relying on collective intelligence and parallel behavior, tries to search for suitable subspaces from the large search space, and this algorithm also needs suitable hardware for efficient implementation.

A method based on genetic refrigeration algorithm is presented to solve the work scheduling problem. This algorithm is classified in the class of population-based heuristic algorithms and thus does not have the problems of traditional systematic methods. In addition, in the proposed method, local search has been used to prevent the algorithm from getting stuck in local optima, and for this reason, this algorithm shows a more intelligent behavior than the genetic algorithm in the conditions of local optima. . In addition, the proposed algorithm does not require special hardware to be implemented and can perform much better than the usual genetic algorithm in terms of time.

3-1- Research methodology

The current research is based on the purpose of the applied type and based on the way of simulating the environment, cloud, data resources and processing and tasks. Considering that it is possible to simulate the cloud computing environment in computer systems, the data required for the research was collected in the computer simulation environment. Most of the necessary data and tools are a computer network and information that is exchanged between the systems in the network. In this regard, firstly, the general structure of the task scheduling model will be introduced using the dynamic adaptive algorithm, then using the genetic algorithm, and finally using the proposed (adaptive-genetic) algorithm, and finally, we will examine the issue that Is it possible to

achieve a suitable and efficient timing method by taking advantage of the special features of the models? Also, MATLAB software was used to analyze the data of the tested cloud environment, and finally the results will be reported using charts and graphs.

The evaluation and testing factors in this research are:

fault tolerance

response time

Efficiency (sum of transferred data)

Related overhead (low-high-medium)

Considering the importance of the load balancing process in cloud computing, the aim of this research is to investigate this process and compare the proposed methods in this field with the help of weight parameters with the genetic refrigeration algorithm approach.

In recent years, optimization algorithms have been well used in all sciences and have been very efficient compared to classical algorithms. On the other hand, work scheduling algorithms, despite their distributable state, lack this intelligence in cloud environments. In this research, we decided to improve the load balance and scheduling speed of tasks in the cloud environment with the genetic cooling algorithm.

In this research, firstly, with the help of graph theory, a suitable mathematical model for allocating work to virtual machines needs to be expressed, and then the cost of allocating work to virtual machines needs to be modeled as a mathematical objective function. After building the cost function, optimal allocation of work and machine will be done with the help of optimization algorithms. It should be noted that the required variables and how to check and measure them are as follows:

Assuming that Tn ... T1 are the tasks requested to a cloud environment and must be executed by machines M1,...,Mm. Also, the aforementioned tasks are related to data sources. R. they need. Each of the tasks has a specific volume and their implementation is completed in a specific time. The scheduling algorithm A must allocate and prioritize machines M and data sources R in such a way that the least possible time is spent on the execution of tasks, the steps of this process (checking and measuring variables) are as follows:

1- Creating a search tree for machines M

2- Selecting n machines with the highest processing power for each task T.

3- Inclusion of the obtained answer in the initial population of the genetic algorithm

4- search for the optimal solution by genetic algorithm

In the following, a brief explanation of the main body of dynamic and genetic adaptive scheduling algorithms as well as their working methods will be examined, and by comparing these two methods, we will find the superiority of each over the other, and in the end, we will combine two dynamic and genetic adaptive algorithms. In parallel with refrigeration, we try to find a better answer than the two mentioned methods.

The main goal of this research is to reduce costs, energy consumption and increase time to increase customer satisfaction and increase the profit of the service provider. Therefore, how to calculate the objective functions is as follows.

The amount of time it takes for a workflow to run from start to finish is known as its lead time. As a result, MKS is determined as follows:

$$\mathrm{MKS} = \max\{\mathrm{FST}_{\mathrm{ti}}, \mathrm{ti} \in \mathrm{T}\} - \min\{\mathrm{SRT}_{\mathrm{ti}}, \mathrm{ti} \in \mathrm{T}\} \quad (1)$$

Here, SRTti and FSTti mean the start and end times of a workflow, respectively.

energy consumption

The active and idle parts of the energy consumption model are denoted by Eac t and Eide, respectively. The term "Eac t" describes the power used while performing a task, and "Eide" refers to the energy consumed when a source is idle. Active energy is determined using it

$$E_{act} = \sum_{i=1}^{n} \alpha f r_i v l_i^2 (FST_{ti} - SRT_{ti}) \quad (2)$$

Here, the supply voltage and frequency for duty sources i are denoted by $v l_i^2 \ f r_i$_in and α is a constant.

The source enters a sleep state during periods of non-use, where the relative frequency and voltage source levels are at their lowest. As a result, the following equation is used to calculate the amount of energy consumed in the inactive state:

$$E_{ide} = \sum_{j=1}^{m} \sum_{idej \in IDEj} \alpha f r_{minj} v l_{minj}^2 LN_j \quad (3)$$

Here $f r_{minj} v l_{minj}^2$as the frequency and minimum supply voltage at source j respectively, and the length of idle time forE_{ide} is denoted by LN_j. The total amount of energy (TE) used by the cloud system to complete The operation is complete

$$TE = E_{act} + E_{ide} \quad (4)$$

Computational cost

There are monetary costs associated with each task a compute node completes. There are two components to the computational cost of a given task: processing cost and memory, which can be estimated as follows.

$$CS_i^{cmp} = \sum_{j=1}^{m} (CS_j^p \times E_{IJ}^{bdw} + CS_j^m \times T_i^{mcm}) \times x_{ij} \forall i \in \{1, \ldots, n\} \quad (5)$$

Here x_ij is zero or one. If cloud and fog nodes are available for task ti, the value is xij , otherwise the value of xij is 0 , and $CS_i^{cmp}$are constants that cost the node RAM and CPU usage, respectively. Nj. The amount of main memory required for Ti work is denoted by T_i^{mcm}. This equation leads to the following definition of the total computational cost TCmp for a set of n tasks:

$$TCmp = \sum_{i=1}^{n} CS_i^{cmp} \quad (6)$$

Communication cost

The communication cost for a particular task is considered in addition to the computational cost. The size of the total output and input work items as well as the bandwidth usage price per node data unit determine this cost. Let CS_j^p be the bandwidth utilization rate of node Nj in data unit and T_i^{bdw}be the required bandwidth of task Ti in bytes.

$$CS_i^{comp} = \sum_{j=1}^{m} (CS_j^p \times T_i^{bdw}) \times x_{ij} \forall i \in \{1, \ldots, n\} \quad (7)$$

Below is how to determine the cost of Ti task communications.

Consequently, the following equation gives the total communication cost for all n tasks.

$$TComu = \sum_{i=1}^{n} CS_i^{comp} \quad (8)$$

total cost

Now we can get the total cost using the following equation.

TCost = TComu + TCmp (9)

Finally, from the above description, the objective function is defined as

Obj = min(MKS + TE + TCost) (10)

3-2 dynamic adaptive scheduling algorithm

The structure of this model is based on the method of retrieving the required information from several data sources, and by providing a dynamic and heuristic schedule, it allocates processor resources to interdependent tasks based on parallel retrieval time from several sources and work processing time. Therefore, with a combination of data retrieval and task mapping technique, the dynamic adaptive model processor can achieve proper scheduling and more efficient than other methods. A scheduling mechanism should be able to select data sources and transfer information from them to processor resources at an optimal time. On the other hand, the computer performing the tasks related to the desired data sources should be selected so that the processing time is minimized. In order to sort the resources in the dynamic adaptive model, the data and processing resources in a nested loop structure are constantly in use and the scheduling algorithm searches among them. Therefore, it seems very necessary to categorize the resources available in the cloud in a suitable and efficient way so that the search can be done in the shortest possible time. Obviously, for a computer system and data center, there are multiple criteria and criteria for classification, such as CPU speed, memory, hard disk driver, power and speed of data transfer in the network (bandwidth) and.... considering that Sorting in this case is not like sorting one-dimensional arrays, the most suitable option is the KD tree.

The logic and heuristic structure of the dynamic adaptive algorithm is as follows: for each new task entered in the waiting queue, the dynamic adaptive algorithm explores computing resources and data centers and obtains the task completion time. The algorithm places all of them in a tree structure to discover the idle or low-working computing nodes faster. A subset of processing resources with the highest degree of connection between each other and the most suitable machine is taken from the tree. If the tasks are roughly the same size, the idle time of the CPU can be considered equivalent to the amount of free RAM space, since the more free RAM space is available, it means less programs to load and less CPU usage. It is Yu. Elements of the R collection are candidate resources for implementing new tasks. For each task i present in the waiting queue, the earliest possible execution time is calculated by the R processing nodes, and the earliest execution time is obtained by considering the end execution time of the current programs loaded in Zem. Start Time is a variable that stores the start time of tasks and is initially considered equal to a large value (StartTime). After that, the time it takes to transfer the data and information required by the tasks to the host processor should be calculated. The bandwidth between the computing host system and the neighboring data center is considered equal to the value by the following relationship:

$$BandWidth = Min \begin{bmatrix} NetSpeed(CompNode) \\ , \\ NetSpeed(AdjacentNode) \end{bmatrix} \quad (11)$$

Figure 1. The process of finding the most suitable processor node for a specific task. Before executing the tasks in the computing host, the information they need must have reached the node executing the tasks. In this article, at first, the machines that have a higher score in terms of resources and also in terms of delay time need to be selected, and at first a proper screening is done with the help of KD tree. For this purpose, a dataset of suitable machines as well as unsuitable machines is defined for the tree, and the tree first learns which machines to choose for the initial population in two phases of training. For this purpose, KD tree is used. The machines selected by the KD tree are suggested to the parallel genetic algorithm for the initial population.

2-4Structure of the proposed genetic refrigeration algorithm.

4-3-1Proposed genetic algorithm

Permutation coding is used in the proposed algorithm. The formation of this encoding consists of two steps: 1) for each machine m_i, a sequence S_i is created from the tasks assigned to that machine, and 2) the sequences of tasks created in the first step are concatenated. The result of these two steps is a permutation of the tasks assigned to the machines. The figure below shows an example of this type of coding for 10 jobs and 5 machines. This coding consists of two parts: 1) the scheduling and assignment of tasks and 2) the data structure related to the display of the number of tasks assigned to each machine.

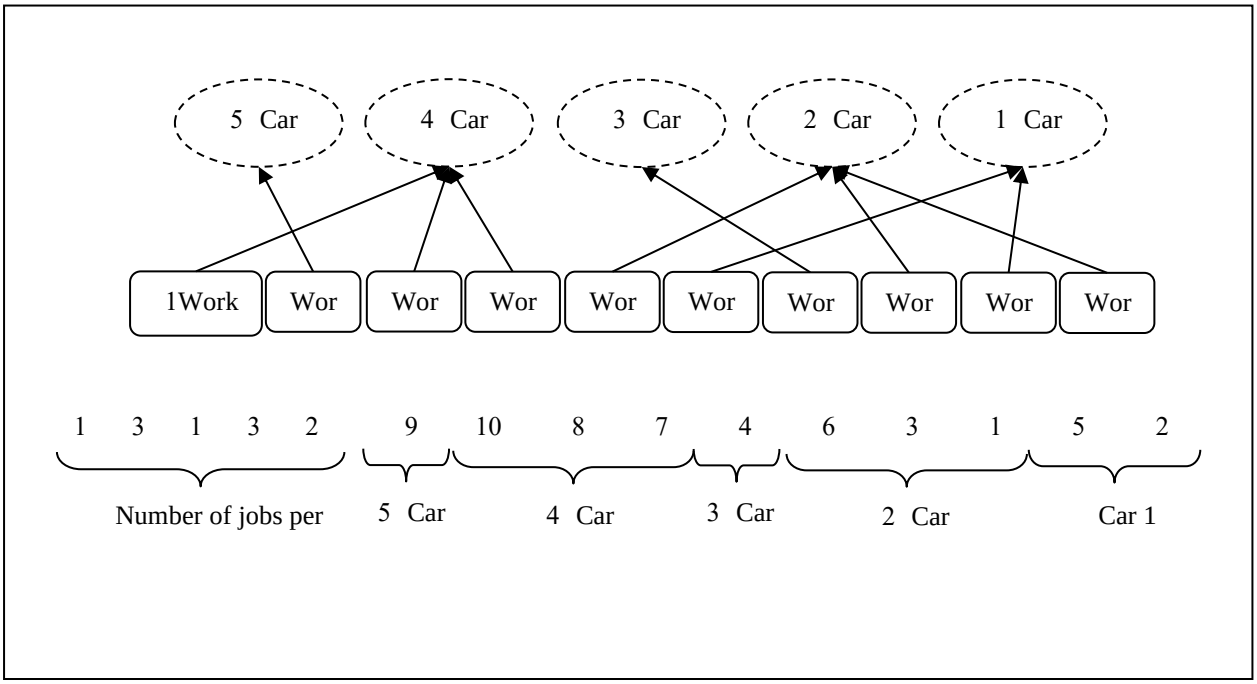


Figure 1: An example of permutation coding for scheduling 10 jobs on 5 machines

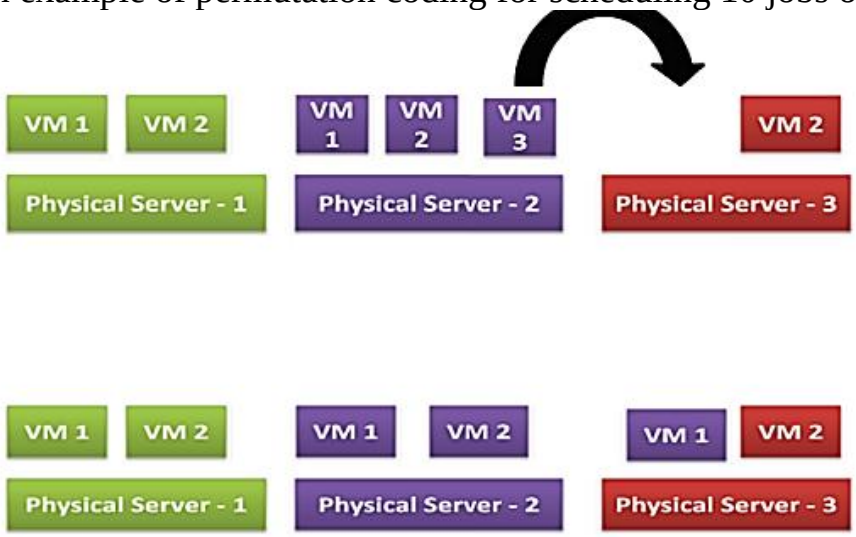


Figure 2: Migration from a car with a heavy load to a car with a low load

The first step in the genetic algorithm is to create an initial population of chromosomes. The generation of the initial population is done in the form of random distribution of processors to working groups. In this method, the diversity of chromosomes is high because the chromosomes belong to different regions of the solution space. As a result, in the initial iterations, the evolution algorithm Generations are done faster with increasing repetition, the similarity of chromosomes also increases until finally the algorithm converges to one or more index solutions, in other words, after randomly allocating resources to the tasks of the initial population, which are entered into the system as groups of 40 tasks.

$$[job - groups] \leftarrow randi\ (cpu) \tag{12}$$

In fact, the primary population is a matrix, the number of rows of which is the population of the group and the number of columns is the number of jobs, and the processors are mapped to it in a random distribution, then each of them is evaluated with each other in terms of the cost function, which is the time to complete the execution of the jobs.

$$F(cost) = \max\{(run\ time)i\} \quad , i \in \{1,2,\dots,n\} \qquad (13)$$

where n is the number of tasks. Then, using the cost function formula, we choose the most suitable parents, and by removing unsuitable parents from suitable parents, we increase the competence of the community of selected parents for reproduction, which improves the distributed genetic algorithm.

From this algorithm, in each working group entered into the problem solving system, there are three populations, including the improved main population, the population of children, and the population of mutants. For this purpose, the chromosomes of these three populations are merged with each other based on the quality of their answers and the objective function of the problem to create a merged population. In other words, the corresponding value of each chromosome is placed in the objective function of the problem, and chromosomes with the best possible answer and with the shortest time to complete the tasks are selected for the final population.

In optimization, the goal is to find an optimal solution according to the variables of the problem. We create an array of problem variables to be optimized.

When applying population-based algorithms such as genetic algorithm and genetic refrigeration algorithm to a problem, two situations may occur, firstly, the algorithm converges towards the global optimal solution, in this case, the efficiency of the algorithm improves over time and the possible solutions It will produce acceptance. Second, if the algorithm converges towards local optima, in this case, the efficiency of the algorithm will be severely destroyed and the generated solutions will not be reliable. Considering that the issue of scheduling and resource allocation in cloud computing is highly non-linear, the possibility of the second scenario is very likely. In the proposed algorithm, to overcome this problem, a method based on local search is used.
The more the genetic algorithm converges to a solution, the similarity between the individuals of the population increases, and therefore the merit values of individuals become more similar and the diversity of the population decreases. The proposed algorithm uses this logic to control the time of repeated hill climbing to the genetic algorithm. In this method, a mechanism based on the similarity coefficient is used to measure the similarity between people in the population.

Parallel genetics is used in the proposed algorithm. For this purpose, a set of answers is generated at the beginning, which is the output of the KD tree algorithm. In the next step, mutation and crossover operations are performed on it. Also, due to the parallel nature of the genetic algorithm, the answer sets are generated in several categories and the best answers are selected, and then the proposed answers obtained from the three mentioned stages make a new population with the help of the combination law of the sa algorithm. At this stage, between the previous answers and the generated answers, it is necessary to decide which populations will be used to produce the new generation, in which case the sa algorithm is used. In the sa population composition rule, the following rule is used to combine the initial population

The refrigeration simulation algorithm is presented mathematically and accurately. The probability of transition from a current state, say s to a new candidate state like s′ is defined by an acceptance probability function P(e,e′,T) where e=E(s) and e′=E(s′) as stated earlier, the function E in the state space indicates the internal energy of the system and T indicates the temperature. The temperature T changes with time. Therefore, according to what was said before, the goal is that the energy of the system is minimal, so the state where E(s) is lower is better than the state where the value of E(s) is higher. Attention in the simulated refrigeration algorithm is that the probability function P must always be positive, even in the conditions that e is smaller than e'. This feature makes the algorithm not stop at the local optimum, which is "worse" than the global optimum. In fact, if s' is better than s E(s)≥E(s'), s' is accepted, and if s' is worse than s and causes E(s)<E(s'), s′ is accepted with one probability.The probability function P is as follows:

$$p = e^{-\frac{E(st)-E(s)}{T}} = e^{-\frac{\Delta f}{T}} \qquad (14)$$

If the temperature T decreases and tends to zero, the probability of P should also decrease and either tend to zero or a positive number. In this situation, P tends to zero when e<e′ and tends to a positive number when e′<e. As can be seen, temperature plays a fundamental role in controlling system changes. As mentioned earlier, the temperature decreases gradually in the simulation. Therefore, the algorithm starts its work from T=∞, in other words, from a very large temperature and decreases in each step according to a predetermined cooling schedule.

Refrigeration scheduling is done according to the fact that if the used resources (for example, the amount of calculations) end, the time of the process also ends. Therefore, the algorithm first looks for an answer in a large space of solutions, regardless of the internal energy of the system, and gradually moves to areas with less energy. This region becomes smaller gradually and this work continues until the global optimum is found. For a problem with a finite space, the increase in time causes the algorithm to stop at the global optimal point with a probability of one; However, the time required to do this is more than the time required to search the entire space, and therefore, it is not practical.

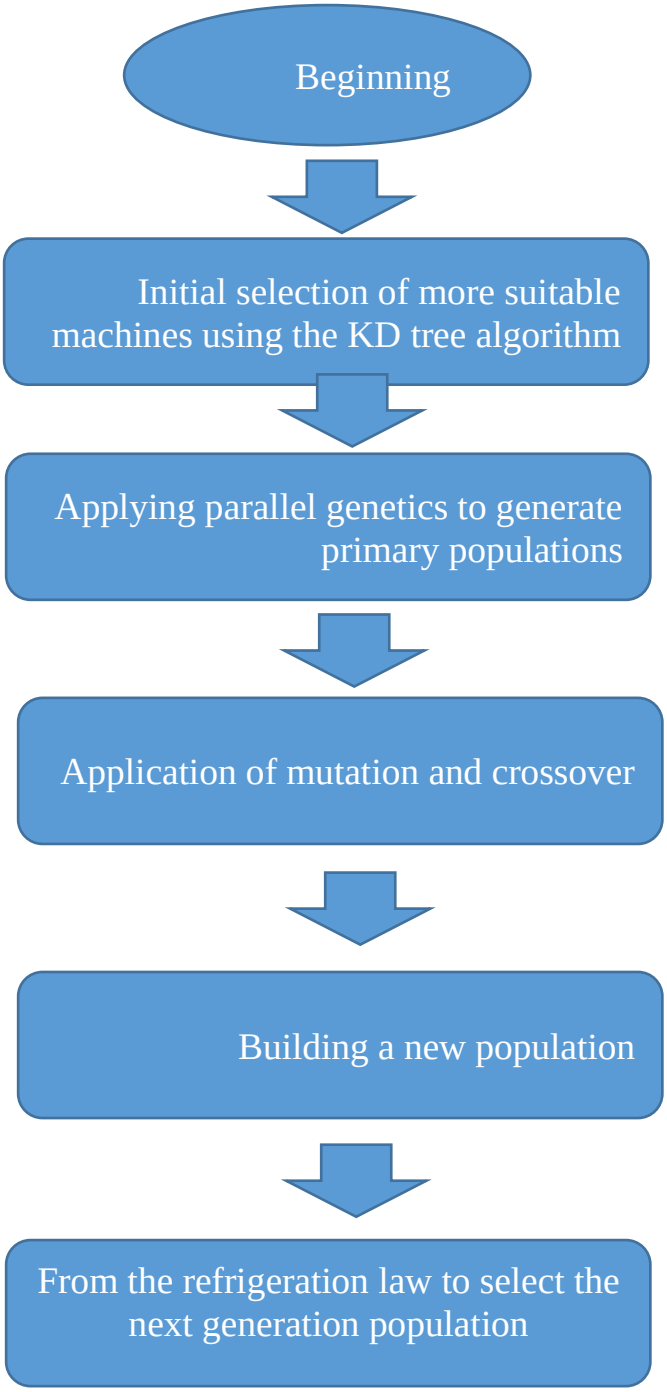


Figure 3: Proposed flowchart

The software used in this research is MATLAB.

5- Simulation scenario

In order to evaluate the proposed simulation algorithm for a cloud environment has been investigated. In the cloud environment, there are 10 virtual machines with the following conditions, each of which has certain characteristics

Machine 1: processing power 90, RAM 80, work in progress 35, energy consumption 75

Machine 2: processing power 50, RAM 44, current work in progress 75, energy consumption 64

Machine 3: processing power 60, RAM 53, work in progress 28, energy consumption 75

Machine 4: processing power 5, RAM 25, work in progress 75, energy consumption 75

Machine 5: processing power 40, RAM 35, current work in progress 96, energy consumption 46

Machine 6: processing power 96, RAM 75, work in progress 67, energy consumption 64

Machine 7: processing power 43, RAM 75, work in progress 97, energy consumption 67

Machine 8: processing power 66, RAM 25, work in progress 97, energy consumption 97

Machine 9: processing power 34, RAM 95, work in progress 88, energy consumption 68

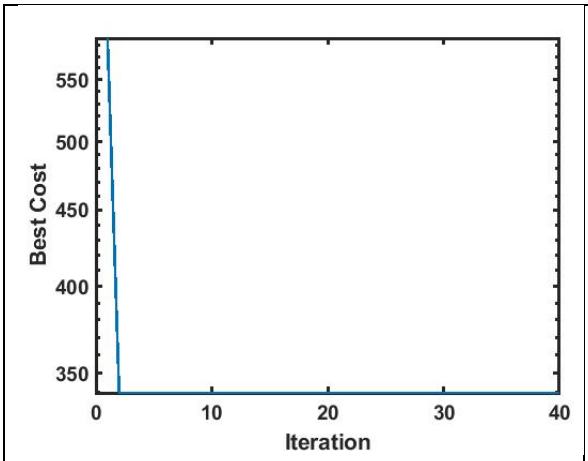
Machine 10: processing power 90, RAM 25, work in progress 56, energy consumption 56

We are going to allocate 8 jobs between virtual machines, for this purpose it is necessary to calculate the cost function for the machines, for this purpose, the total cost function for the machines calculated as follows is considered, and the machine and work respectively which create the lowest cost is recognized as the optimal combination of the algorithm.

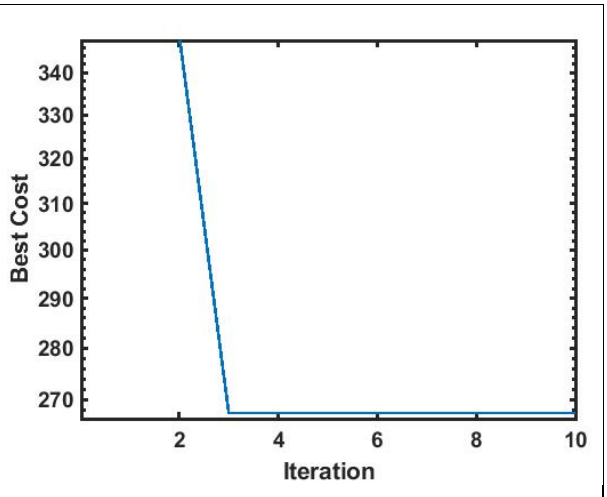
The cost function is considered as follows:

$$cost = \sum(energy + ram + processing) \quad (15)$$

In this article, firstly, in order to identify the suitable machines and select the suitable initial population from among the introduced machines, suitable machines are classified with the help of tree classification and are provided as the initial population to the meta-processing algorithm. In the next stage of the meta-processing algorithm Genetic refrigeration along with genetic algorithm has been used to identify and optimally allocate work and machine, for this purpose the following code has been used. The results are shown for the genetic algorithm and the genetic algorithm of genetic refrigeration



Optimization using pso algorithm



Optimization using the proposed algorithm

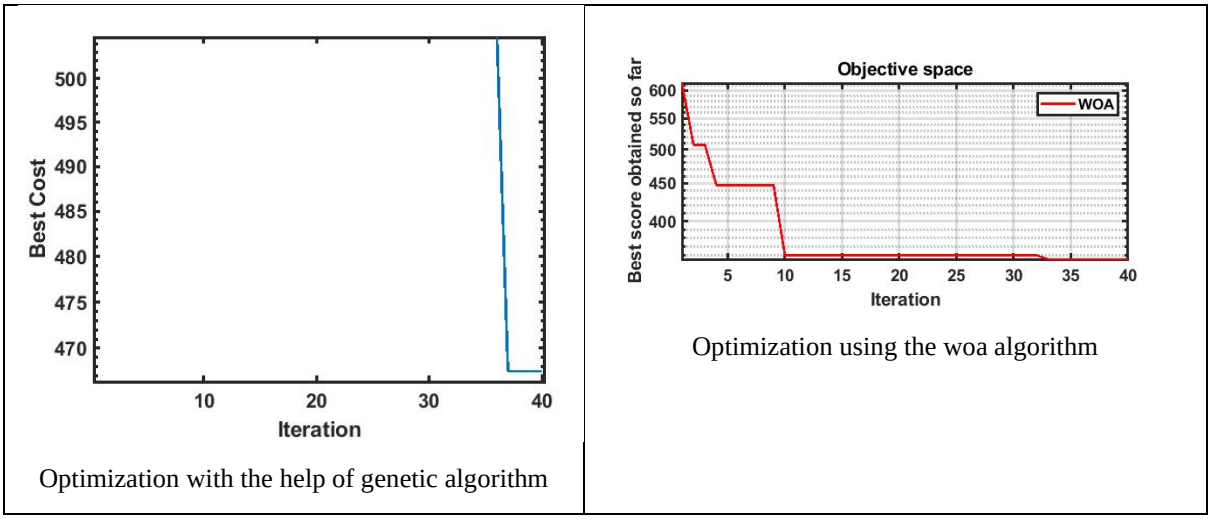


Figure 4: Comparison of the results between genetic algorithm and genetic refrigeration genetics

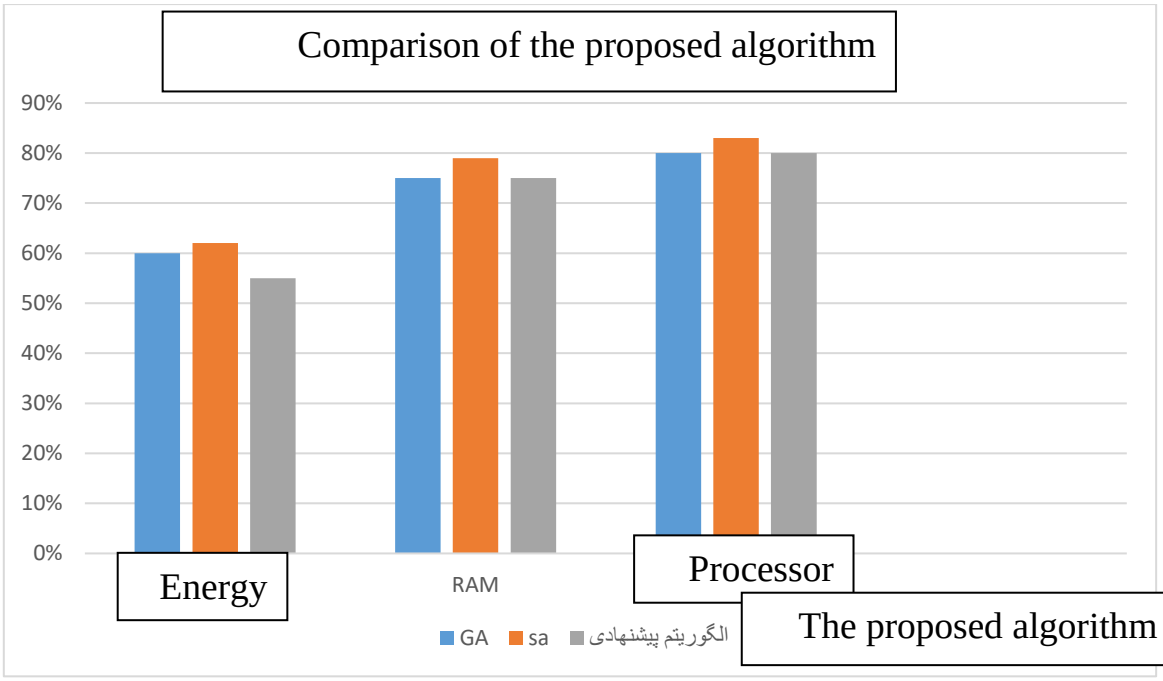


Figure 5: Comparison of results and sources

6-Conclusion

The simulation of the proposed algorithm has been done in a scenario with a certain number of virtual machines and the work has been done. For this purpose, 10 virtual machines with different specifications have been considered. Eight tasks with the same indicators have been selected to be assigned to these 10 machines. For this purpose, the genetic refrigeration algorithm has been compared, the problem of resource allocation in cloud computing and the scheduling of each of the user's tasks on virtual machines in a cloud. It is an NP-Complete problem that many algorithms have been proposed to solve. In this study, the combined algorithm of genetic refrigeration algorithm and local search was presented as a proposed algorithm, and the quality of the answers and its efficiency were compared with the genetic algorithm. According to the results obtained from the simulation of these algorithms, it was found that the proposed algorithm works faster than the genetic algorithm in terms of execution time quality. In addition, it was found that the proposed algorithm has performed better than the other compared algorithms in terms of the quality of the answers. In other words, the schedules produced by this algorithm have lower costs.

The scheduling mechanism can meet the requirements of users who use the cloud environment, considering the fact that the scheduling systems that have been provided so far for cloud processing have not been very successful due to the low level of accuracy of the provided algorithms. We have designed and implemented a new scheduling system with evolutionary algorithms combined with genetic algorithms so that we can improve the scheduling problem in dependent tasks to a great extent. Among these proposed hybrid algorithms, the GA_SA algorithm has shown impressive performance. In addition to being successful in improving the completion time of tasks in the cloud system, this algorithm has also been shown to be better than other evolutionary algorithms in the time flow of tasks. Another feature of this algorithm is to have a suitable speed to find the global optimum.

Resources

[1] de Oliveira-GRR20112021 FAN, Risso-GRR20120726 JVT. Dynamic resource allocation in software defined and virtual networks: a comparative analysis.

[2] Bari MF, Boutaba R, Esteves Ret al. Data center network virtualization: a survey. IEEE Commun Surv Tutorials. 2013;15(2):909–928.

[3] Endo PT, Batista MS, Goncalves GE, et al. Selforganizing strategies for resource management in Cloud Computing: State-of-the-art and challenges. 2nd IEEE Latin American Conference on Cloud Computing and Communications (LatinCloud); Maceio, Brazil; 2013.

[4] Nguyen Van H, Dang Tran F, Menaud J-M. Autonomic virtual resource management for service hosting platforms. Proceedings of the ICSE Workshop on Software Engineering Challenges of Cloud Computing; IEEE Computer Society; Vancouver, Canada; 2009.

[5] Ullrich M, Lässig J, Gaedke M. Towards efficient resource management in cloud computing: a survey. IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud); Vienna, Austria; 2016.

[6] Mell P, Grance T. The NIST definition of cloud computing; 2011.

[7] Rahmani, O., Taherkhani, A., Rahmani, M., & Karimiyan, T. (2014). The optimal design of earthing system based on genetic algorithm. Advances in Environmental Biology, 644-657.

[8] Zamani, M. , Azadi, S. , & Rahmani Seryasat, O. (2021). Noise Reduction in Medical X-Ray Images Using Wavelet and Neural Networks. Transactions on Machine Intelligence, 4(1), 36-52. doi: 10.47176/TMI.2021.36

[9] Vares, H. , Sabzehparvar, M. , & Bannazadeh, M. J. (2021). Using a Genetic Algorithm, Integrated Preventive Maintenance Planning and Production Scheduling for a Single Machine. Transactions on Data Analysis in Social Science, 3(1), 40-45. doi: 10.47176/TDASS/2021.40

[10] Lee G, Katz RH. Heterogeneity-aware resource allocation and scheduling in the cloud. HotCloud; 2011.

[11] Sangeetha, S.B., et al., Resource Management Framework Using Deep Neural Networks in Multi-Cloud Environment, in Operationalizing Multi-Cloud Environments. 2022, Springer. p. 89-104.

[12] Abualigah, L., & Alkhrabsheh, M. (2022). Amended hybrid multi-verse optimizer with genetic algorithm for solving task scheduling problem in cloud computing. The Journal of Supercomputing, 78(1), 740-765.

[13] Koneru S, Uddandi VR, Kavuri S. Resource allocation method using scheduling methods for parallel data processing in cloud. Int J Comput Sci Inf Technol [IJCSIT]. 2012;3(4):4625–4628.

[14] Maguluri ST, Srikant R, Ying L. Stochastic models of load balancing and scheduling in cloud computing clusters. International Conference on Computer Communications; Orlando, FL; 2012.

[15] Kang Z, Wang H. A novel approach to allocate cloud resource with different performance traits. IEEE International Conference on Services Computing (SCC); Santa Clara, CA; 2013.

[16] Zhang Q, Cheng L, Boutaba R. Cloud computing: stateof-the-art and research challenges. J Internet Services Appl. 2010;1(1):7–18.

[17] Shi W, et al. An online auction framework for dynamic resource provisioning in cloud computing. IEEE/ACM Trans Network. 2016;24(4):2060–2073.

[18] Rodriguez, MA, Buyya R. Deadline based resource provisioningand scheduling algorithm for scientific workflows on clouds. Cloud Comput. 2014;2(2):222–235.

[19] Basahel, S. B., & Yamin, M. (2022, March). A Novel Genetic Algorithm for Efficient Task Scheduling in Cloud Environment. In 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 30-34). IEEE

[20] Raman, N., Wahab, A. B., & Chandrasekaran, S. (2021). Computation of workflow scheduling using backpropagation neural network in cloud computing: a virtual machine placement approach. The Journal of Supercomputing, 77(9), 9454-9473.

[21] Paulraj, D., Sethukarasi, T., Neelakandan, S., Prakash, M., & Baburaj, E. (2023). An Efficient Hybrid Job Scheduling Optimization (EHJSO) approach to enhance resource search using Cuckoo and Grey Wolf Job Optimization for cloud environment. Plos one, 18(3), e0282600.

[22] Singh A, Juneja D, Malhotra M. A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. J King Saud University Comput Inf Sci; 2017;29(1):19–28.

[23] Javadi B, Abawajy J, Buyya R. Failure-aware resource provisioning for hybrid Cloud infrastructure. J Parallel Distrib Comput. 2012;72(10):1318–1331.

[24] Kaur P, Mehta S. Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. J Parallel Distrib Comput. 2017;101:41–50.

[25] Yang S, et al. Energy-aware provisioning in optical cloud networks. Comput Networks. 2017;118:78–95.

[26] Cheraghlou MN, Khadem-Zadeh A, Haghparast M. A survey of fault tolerance architecture in cloud computing. J Network Comput Appl. 2016;61:81–92.

[27] Ghobaei-Arani M, Jabbehdari S, Pourmina MA. An autonomic resource provisioning approach for servicebased cloud applications: a hybrid approach. Future Gener Comput Syst. 2017.