Sustainable Computing: Informatics and Systems 28 (2020) 100431

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

journal homepage: www.elsevier.com/locate/suscom

Power efficient resource provisioning for cloud infrastructure using bio-inspired artificial neural network model

Pradeep Singh Rawat [a], Punit Gupta [b,*], Priti Dimri [c], G.P. Saroha [d]

[a] Uttarakhand Technical University, Dehradun, India
[b] Manipal University Jaipur, India
[c] Govind Balabh Pant Engineering College, Pauri Garhwal, India
[d] Maharshi Dayanand University Rohtak, India

ARTICLE INFO

Keywords:
ACO
AFs
ANN
Big-Bang Big-Crunch
FCFS
LReLU
Scheduling
Power

ABSTRACT

High-performance computing changes the way of computing. More than one-decade, the cloud computing paradigm has changed the way of computing, communication, and technology. The efficient resource provisioning or task scheduler policy improvement is a challenging issue in the service-oriented computing paradigm. This article work focuses on task scheduler policy improvement for better cloud application performance. The task scheduling algorithm aims to improve the performance of real-time applications in the cloud by reducing task waiting time, execution time, and power consumption. The proposed model is inspired by an Artificial Neural Network (ANN) based system with a training model using a genetic algorithm. Results exhibit that the proposed GA-ANN policy outperforms the (Big-Bang Big-Crunch cost-aware), Genetic cost-aware, and other existing approaches. The results show that the proposed GA-ANN model performs better than existing approaches taking power consumption, total completion time(ms), average start time (ms), and average completion time(ms) as performance metrics. The proposed GA-ANN model is validated using real-time user requests (standard workload file) from workload traces (San Diego Supercomputer Center (SDSC) Blue Horizon logs), and fabricated data sets. The proposed model improves power efficiency by 13 %, scheduling time by 77.14 % and total execution time by 36 %. Hence the proposed GA-ANN technique provides performance as compared to existing systems.

1. Introduction

The scalable cloud computing environment upgrades the benefit from the data center resources. The objective function achieves using distributed elastic resources which depends on the quality of service enhancement metrics. The service-oriented computing paradigm provides scalable infrastructure, platform to run the applications. The resources show their availability at the host level inside the data center. Consumer higher the resources using pricing models. It includes a subscription-based pricing model of data center components and provides massive IT resources as a service using a high-speed network [1]. Service level agreement is the legal contract between cloud service providers and cloud computing customers, which assures the resource availability with minimum violation of service agreement. Task scheduling on virtual machines is the primary concern in a scalable cloud environment [2]. Cloud service providers assure the minimum outage time of cloud infrastructure. Task scheduler maps the users' requests on an appropriate resource (virtual machine) with high processing power. The Cloud computing paradigm provides shared resources with rapid elasticity to allocate the resources with the minimum operational cost of the resources. It follows the features of utility computing, electricity, and power supply. Emerging of cloud computing moves the computing power and data from traditional computing architecture to the scalable shared pool of resources. End-users access and use all the services without any prior information about the physical location of the infrastructure, platform, and network resources. The scheduling of the task focuses on finding an optimal solution globally. The researches focused on the state of art methods for the cloud resource provisioning concern using static, dynamic, and meta-heuristic techniques. In [3] authors presented artificial Intelligence techniques that deal with challenging issues in cloud resource optimization. Artificial intelligence techniques deal with the challenges in the cloud computing environment i.e.

* Corresponding author.
E-mail address: punitg07@gmail.com (P. Gupta).

https://doi.org/10.1016/j.suscom.2020.100431
Received 29 January 2020; Received in revised form 9 June 2020; Accepted 13 August 2020
Available online 18 November 2020

efficient resource allocation, data center management, power-aware scheduling, and fault aware scheduling. Our focus point includes the power-efficient scheduling in scalable cloud aura. The nature-inspired artificial intelligence base, Genetic algorithm, colony optimization, and Big-Bang Big-Crunch cost-aware approaches are used for solving SaaS modeler scheduling on the IaaS cloud model. Researchers focused on energy saving and response time analysis using heuristic-based approaches. The performance can be improved using a hybrid approach based on bio-nature inspired and artificial neural computing. Our primary focus includes the Genetic-ANN model which may perform better than simple bio-inspired Genetic technique using real and fabricated data sets. Bio-inspired and nature-inspired techniques provide an optimal global solution [4].

In [5] authors proposed a technique using the ANN model. The objective function covered the energy-saving and minimization of makespan. The proposed ANN-based technique predicts the virtual machine for mapping the tasks which provide an optimal solution. The proposed ANN-based model is trained using data sets i.e. fabricated or real data sets. In [6] authors proposed a bio-inspired modified genetic approach. The primary focus includes the optimal scheduling of the tasks. Performance is measured using completion time, response time, and quality of service parameters. Hence authors addressed the performance issue of the cloud system using bio-inspired scheduling. Hence motivation of the research work includes the performance gain using the proposed Neuro-Bio inspired technique in a scalable cloud era. The quality of service improves using optimization metrics, power consumption (Kwh), execution time, simulation time, makespan, and average start time (ms). Section 5, covered the performance metrics results using GA-ANN and existing resource provisioning techniques. The simulation is performed using Cloudsim 3.0 using real and fabricated data sets. Section 5, presents the simulation results using real data sets and fabricated data sets.

The key contributions of our proposed technique can be summarized as follows: Authors proposed a power-efficient resource provisioning for Cloud Infrastructure. The proposed model is based on Bio-Inspired Artificial Neural Network Model. The bio-inspired neural computing model takes the cloud tasks and set of virtual machines as an input parameter in a scalable cloud computing environment. Our proposed power efficient model focus on performance metrics power consumption, total completion time, average start time and average completion time, and server utilization. The objective function also depends on resource utilization cost

1.1. Organization of the paper

Section 2 covers the related work done in meta-heuristic algorithms and energy-aware scheduling in a scalable cloud scenario. Section 3 illustrates the motivation of the work. Section 4 describes the proposed task scheduler policy using ANN (Artificial Neural Network) model and genetic algorithm. Section 5 illustrates the simulation environment and the results of the proposed model using real-time workload data sets and fabricated data sets. Finally, the article is concluded in section 6.

2. Related works

Efficient task provisioning is a challenging concern in the IaaS cloud. The related works include the detailed literature review on static, dynamic, and nature-inspired meta-heuristic approaches. The authors presented the multi-objective meta-heuristic technique Ant Colony optimization. The proposed model focused on performance metrics, resource utilization, service reliability, load balancing, and network transmission overhead are the key focusing parameters [7]. In this article, the authors focused on the virtual machine placement model which improves the utilization of the servers and improves the power saving. The author also covered that the evolutionary computing model improves optimization metrics [8]. In this article authors proposed two energy-aware scheduling algorithms which included energy-aware scheduling and energy-aware migration. Energy-aware model proclaimed better efficiency than the existing provisioning technique includes First Come First Serve (FCFS), local probing techniques like Stochastic Hill Climbing (SHC), bio-inspired Genetic Algorithm (GA). The author focused only on limited performance evaluation criteria without using the artificial neural network model [9].

In [10] authors proclaimed an enhanced nature-inspired Ant algorithm for the scheduling using multi-objective optimization parameters. The optimization metrics include makespan, cost, deadline violation rate, and resource utilization. The unutilized time of the resources and makespan parameters are used for performance evaluation. The authors also focused on an efficient cost model for resource utilization usages. In [11] authors presented a green cloud provisioning model in the cloud era. The performance metrics include energy-saving and deadline constraints. Cloudsim toolkit uses for simulation purposes. In [12] Authors presented a comprehensive survey on scheduling in the cloud computing environment. The authors focused on the methodical analysis of resource management. The review work will be helpful in our proposed neuro bio-inspired ANN model. In [13] authors presented a hybrid technique based on nature-inspired bio and ant-based scheduling. Authors focused on three performance metrics i.e completion time, response time, and throughput. The proposed hybrid approach may be helpful for future research direction to improve the quality of service using more performance metrics. In [14] authors presented the dynamic approach for resource allocation using optimization metrics improved task allocation and power saving. Further, the performance improves using a nature-inspired hybrid technique.

In [15] the author studied the survey on intelligent scheduling techniques in cloud computing. The survey covered the intelligent and expert scheduling which may helpful for future directions of research. In [16] author proposed bio-inspired scheduling in a grid computing environment. A grid scheduling environment may extend in service-oriented cloud computing. In [17] authors studied the review of optimized provisioning techniques to achieve an optimization in cloud scheduling. The author investigated the bio-inspired technique and swarm intelligence based scheduling techniques. It may helpful for extending the research in scheduling. The author presented the nature-inspired evolution technique for a better solution to scheduling. The optimization metrics makespan, throughput, and completion time are considered to measure the performance [18]. In [19] authors presented through the analysis of nature-inspired particle swarm intelligence technique. The primary focus included the workflow base scheduling in a scalable cloud environment. The research may help to extend the nature-inspired technique using neural computing. In [20] authors proposed a virtual machine intelligent scheduling strategy based on a machine learning algorithm. The primary objective of the proposed model includes balancing the load at the cloud datacenter node. The performance metrics include minimum migration costs. It includes the bio-inspired approach for quality of service improvement. In [21,22] authors focuses on broker level policy implementation for quality of service. Performance is measured using load balancing with minimum response time and datacenter processing time. In [23,24] authors have proposed a scheduling methodology using machine learning techniques to improve the performance of cloud services. The proposed algorithms are learning based algorithm which will work for all type of environments i.e. under loaded and over loaded conditions.

Sheikh H. F. et al. [25] focused on resource provisioning techniques using optimization metrics, power consumption. The performance is evaluated using single and multiple processing elements. The authors also covered a comprehensive survey of power-aware scheduling techniques. Sheikh H. F. et al. [26] presented an in-depth survey of the thermal aware tasks scheduling techniques in a system having multiple processing elements at the host level. The authors showed some observations that may be helpful for the new direction of the optimization objectives in a scalable cloud scenario. Sheikh H. F. et al. [27] proposed

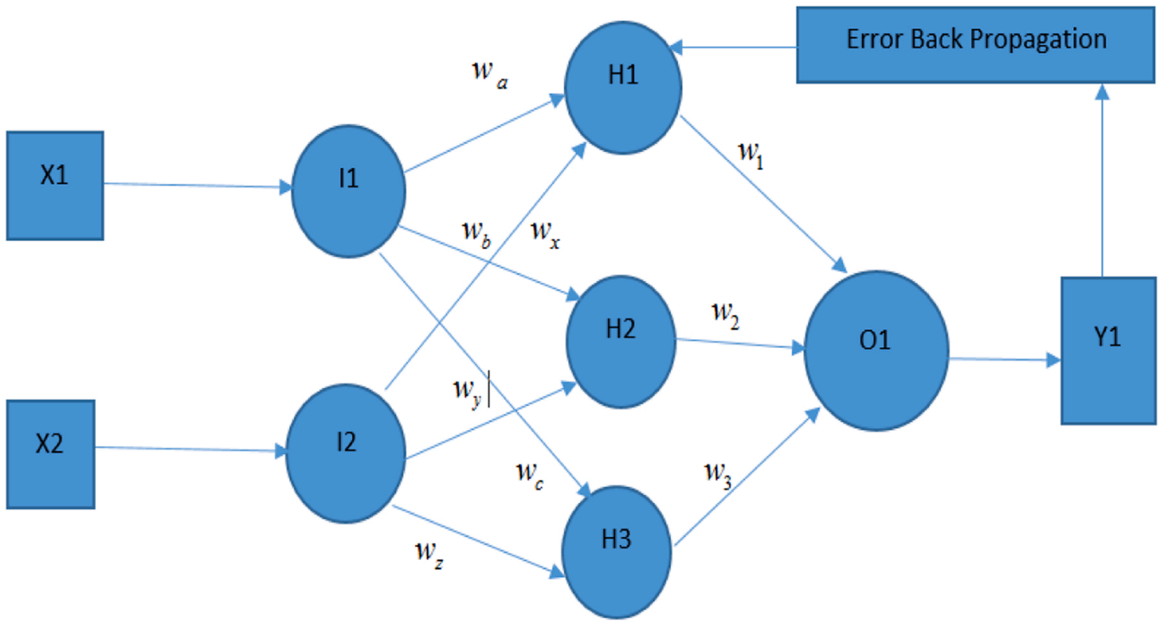


Fig. 1. Two-Layer perceptron model for Resource Provisioning in Scalable Cloud environment.

a multi-objective evolutionary technique using two optimization metrics i.e. energy and temperature. The authors also focused on two optimization metrics without any isolation. The genetic bio model is considered as a base. Hence the literature review exhibit that power, temperature parameters are considered. In [29] authors presented the energy efficient independent tasks scheduler using performance metrics makespan, energy consumption, execution overhead. The primary focus of the authors covers green datacenter in a scalable cloud environment. The proposed technique is based supervised neural network model. In [30] authors presented the time aware and energy-aware technique for task scheduling in a heterogeneous environment. The proposed technique may helpful for new research directions to propose a new hybrid approach based on the meta-heuristic and neural computing model.

In [32] authors proposed a novel technique to map the tasks on virtual machines, and an artificial neural network is used for the future predication of the spot instances. The performance is measured using performance metrics time, cost, and reliability. In [33] authors developed a two-stage strategy for dynamic scheduling of tasks on virtual machines. The primary focus included the time saving of virtual machine creation dynamically. In [34] authors proposed an independent task scheduler using a supervised neural network model. The efficiency is measured using energy, makespan, execution overhead, and the number of active racks inside the datacenter. L. Bao et al. [35] developed a microservice-based application workflow scheduling which supports the scalability feature. The performance is measured using the end to end delay parameter. S. Guo et al. [36] proposed a dynamic resource scheduling and offloading technique. The performance is measured using power consumption and completion time. P. Kuang et al. [37] considered the power-efficient scheduling policy using physical machine status and virtual machine migration process.

S. Pang et al. [38] developed a bio-inspired hybrid scheduling approach that assigns the tasks on a virtual machine. The performance is measured using load balancing ability and task completion time. T.A.L. Genez et al. [39] presented a workflow scheduler in a hybrid cloud using performance metrics makespan, cost, and bandwidth. H.R. Faragardi et al. [40] presented a novel resource provisioning mechanism for workflow scheduling. The performance is measured using performance metrics makespan on the IaaS cloud. Q. Wu et al. [41] modeled the multi-objective evolution based scheduling technique and focused on simultaneous optimization of makespan and economical cost. V. Arabnejad et al. [42] introduced the budget deadline aware scheduling technique. The authors also studied the novelty of the proposed technique using sensitivity analysis. S.K. Mishra et al. [43] focused on two levels of provisioning in the scalable cloud scenario. The provisioning covered tasks to virtual machine mapping and virtual machine to host mapping. The performance is measured using power consumption. M.K. Gupta et al. [44] presented a multi-objective virtual machine placement for IaaS Cloud. The proposed model minimizes the performance metrics power consumption, service level agreement, and improve resource utilization. N. Mansouri and M.M. Javidi [45] proposed a job type and cost-based job scheduling technique. The performance is measured using the response time of the data-intensive and computing-intensive jobs. Y. Lu and N. Sun [46] proposed a resource-aware load balancing based task scheduling technique. The optimization metrics included i.e. power-saving and resource utilization. The proposed technique supports the green computing environment. M.R. Shirani and F. Safi-Esfahani [47] presented a dragonfly dynamic scheduling technique. BBO-Mexican hat wavelet-dragonfly dynamic scheduling framework is used for efficiency improvement. The performance metrics included i.e. execution time, response time, and reduction of service- level agreement violation.

Our primary objective focuses on power saving, time, and costs trade-off in a cloud computing environment using a constant IaaS cloud. Researchers focused only on meta-heuristic techniques and their performance metrics. Our motivations include the convergence rate improvement of task allocation techniques. The convergence rate and optimization metrics improve using a bio-inspired technique. The bio-inspired analogy includes the SaaS modeler in high-performance computing. It covers a single perceptron, and multi perceptron layer neural network model. This is known as an artificial neural network. Artificial Neural network has wide application in performance improvement of the scalable cloud aura. The prominent features of the ANN model are inherited in a high-performance computing paradigm. It covers Utility, Grid, Cloud, and distributed computing environment. The high-performance computing invents put the demand of the artificial neural network. The performance of the bio-inspired technique improves using ANN (Artificial Neural Network). In this section, a detailed review has been covered. It discusses the existing techniques for tasks mapping on virtual machines and various parameters are used for quality of service measurement. The nature-inspired meta-heuristic

Table 1
Performance Metrics.

SL. No.	Meta-Heuristic Technique/ Static/Dynamic	Performance Metrics for Task Scheduler Policy	Data Sets 1	Data Sets 2 [28] [31]
1	BB-BC-Cost Aware	1 Total Completion Time (ms) /Makespan / Total Execution Time (ms) 2 Total Time (ms) 3 Utilization of Each Virtual Machine 4 Average Utilization of VM 5 Average Start Time (ms) 6 Average Finish Time (ms)	Fabricated Data Sets	Real Workload (Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs))
2	GA-Cost Aware			Real Workload (Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs))
3	GA-Execution Time Aware			Real Workload (Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs))
4	Neural-GA			Real Workload (Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs))

techniques are reviewed which focus on performance metrics, power saving, response time, temperature awareness using data sets. This section also reveals the disadvantages of existing methods that are to find a global solution from the algorithm like genetic and big bank big crunch algorithm is costly in terms of scheduling time which keeps on increasing with the increase in problem size. To overcome this a neural network-based model is proposed. Our proposed work, a genetic algorithm (meta-heuristic approach) using ANN used to train the network which works better than simple meta-heuristic techniques.

3. Motivation for work

The primary focus includes the performance gain of the meta-heuristic, static, and dynamic task scheduler policies. The quality of service is measured using a scalable cloud. In the reviewed articles, we found that authors focused on static, dynamic, and nature-inspired and astrology based techniques. The state of art techniques provides the direction to develop a model based on neural computing and bio-inspired techniques. The reviewed tasks scheduling technique provides an avenue for a hybrid technique. It helps in developing a scheduling approach that improves the performance of the bio-inspired meta-heuristic technique. The Genetic algorithm is integrated with the artificial neural network model. Hence Fig. 1 exhibits the two-layer perceptron model for the performance improvement of the bio-inspired meta-heuristic approach. The meta-heuristic techniques with artificial neural network (ANN) focus on performance measurement parameters. The performance parameters are exhibited in Table 1. It includes time aware and energy-aware performance metrics. The performance metrics claim the reliability and global optimality of the ANN-based bio-inspired model. The optimal solution measures using the parameters with input layer data sets from real workload (SDSC-BLUE-2000-4.swf) logs. The original workload data sets contain the logs of the San Diego Supercomputer Center (SDSC) Blue Horizon)), and fabricated data sets are also used for modeling and simulation in scalable cloud aura. Simple static, dynamic, and bio-inspired approaches do not provide the optimal solution with increasing load or number of tasks.

The performance metrics are measured using fabricated data sets and real workload data sets. The fabricated datasets generate using configuration parameters of the virtual machines and cloud tasks. The configuration parameters define the size of the tasks and the processing power of the virtual machines. The Standard Workload Format (SWF) is a single format which store and exchange San Diego Supercomputer Center (SDSC) Blue Horizon logs that is used in Parallel Workload Archive [31]. The SWF format contains the job submission and evaluation details on the San Diego Supercomputer Center node. The log contains the fields information about the job submission times, required processing elements, start times, and end times. These fields are used in the proposed GA-ANN model. The logs are imported with SWF files using programing abstraction in NetBeans IDE integrated with Cloudsim 3.0.

4. Proposed model

4.1. Proposed technique based on ANN scheduler

In this article, we introduce a genetic and artificial neural network-based hybrid technique. The review of the existing work in cloud job scheduling use meta-heuristic techniques. The authors presented cloud resource scheduling using different techniques. Most of the researchers use artificial intelligence techniques such as genetic algorithm and ant colony and particle swarm optimization techniques to solve the problem and to find a solution. The solution may converge at a local or global optimal point. Still, there is an opportunity to improve the performance in this research area using performance metrics time and power consumption. Nature-inspired population-based evolution techniques serve a local optimal and optimal global solution. The performance metrics improve further using a meta-heuristic technique with human brain computation or artificial neural networks. Therefore, we propose a three-layer artificial neural network to optimize task scheduling results. The hybrid technique is proposed in which the ANN model is integrated with the Genetic approach. The artificial neural network model accompanies three steps, which include the training mechanism, validation of the trained network model, and prediction is performed using test data sets. The test data sets are the subsets of the training data sets. The training mechanism plays a prominent role in the accuracy, stability, and correctness of the ANN model. The training mechanism includes the error correction based mechanism and memory-based mechanism. The proposed model divides the total task into two parts 20 % and 80 %. The first part will be used for generating a training dataset. The whole complete tasks will be scheduled using a trained neural network model that is trained using a genetic algorithm.

The proposed model aims to improve power efficiency, makespan, and scheduling delay. The artificial neural network with meta-heuristic approaches provides better results than standalone nature-inspired meta-heuristic techniques as shown in the result section. Perceptron and their connection strength (edge weight), and bias values adjust for input data sets. Mutation and fitness selection is performed for the development of the specific schedule. The trained neural network used for the prediction related to the various non-linear applications. The legitimate output generates at the output layer at the end of the learning or training process. Better results achieve using the ANN model with the bio-inspired meta-heuristic approach. Researches put effort into solving the problem of job scheduling in scalable cloud aura. Most of them have used a genetic algorithm, ant colony, and Big-Bang Big-Crunch cost-aware methodology, which solve the problem of virtual machine allocation and find an optimal assignment of the virtual machines. Our objective is to improve the performance of bio-inspired meta-heuristics techniques using the neural network.

The proposed algorithm uses the tasks list and VM list as an input to the model with basic parameters to the model. The proposed model is divided into the following phases:

1 Initialization
2 Training dataset preparation

Table 2
Parameters of the Artificial Neural Network model.

SL. NO.	Variable Type	Perceptron Parameters (Unknown variables)	Known variables
1	Weight	$w_a, w_b, w_c, w_x, w_y, w_z, w_1, w_2, w_3$	INPUT: X_1, X_2
2	Bias	b_1, b_2, b_3	TARGET OUTPUT: Y_1

3 Neural model designing
4 Model training
5 Error Backpropagation and correction
6 Task scheduling.
7 Initialization

In this phase, all the parameters like the number of input layers, hidden layer, and output layer are defined with several neurons in each layer, Learning Rate, Mutation Rate, Population Size, Evolution, Neurons Input Layer, Neurons Output Layer, Neurons Hidden Layer, Activation Function. This phase defines the efficiency and accuracy of the proposed model using these parameters.

1 Training dataset preparation

In this Phase, the dataset for the training phase is prepared using a genetic algorithm with utilization and execution time as a fitness function to improve power efficiency and makespan at the same time. This phase takes a list of tasks, which is a heterogeneous combination of a variety of tasks along with a list of virtual machines available as resources. The input is given to the proposed genetic algorithm.

$$F(x) = \alpha * Utilization + \beta * Total\_Execution\_time \tag{1}$$

Where α+β = 1

$$Total\_Execution\_time = \sum_{task\_i=1}^{n} \frac{Task\_Length_i}{MIPS_j} \tag{2}$$

Eq. 1, denotes the linear weight function with weight factor α for utilization and weight factor β for execution time respectively. The execution time measures using Eq. 2. Where $MIPS_j$ denotes the processing power of the j^{th} virtual machine and $Task\_Length_i$ denotes the length of the i^{th} task. Where 'n' is the number of tasks and j in the allocated VM to i^{th} task. Utilization is the average utilization of all VM's. The output of this phase is a predicted schedule with the least execution time and power consumption. The outcome is a combination of tasks and the VM id allocated to it. The list is divided into two parts of 80 % and 20 % each. Where the first part is given for network model training and 20 percent is given for testing the trained model and error correction using backpropagation.

3 Neural model designing

This phase a neural network is designed using parameters for the initialization phase. The model consists of the input layer, hidden layer, and output layer with activation function an input later and hidden layer to adjust the weight in such a way that the output of the network is similar to the expected output of the training dataset. Fig. 1 shows the two-layer artificial neural network. The output of an adaptive two-layer perceptron model presents the virtual machine identity which handles the user requests. It uses a feed-forward and backpropagation method for achieving the target values. The three-layer artificial neural network-based model improves the performance of the bio-inspired genetic algorithm. The performance of the ANN with bio-inspired technique measures against the Big-Bang Big-Crunch cost-aware, Genetic cost-aware, and Genetic execution time aware techniques. The artificial neural network trains using different learning rates. The accuracy of the output depends on training parameters. The network having several layers and several nodes in each layer. The performance is affected by the training mechanism.

Table 2 shows, synaptic weight is an unknown parameter that is adjusted using a comparison between target output and desired output. The learning mechanism defines the connection strength between neurons from layers in a multi-layer perceptron model. The mathematical and computational representation of the proposed ANN technique schedules the tasks in scalable cloud aura. Eq. 3 exhibits the output using an activation function.

$$NN(X_1, X_2) = Relue(w_1 * H_1 + w_2 * H_2 + w_3 * H_3 + b_3) \tag{3}$$

WhereH_1,H_2,H_3 measures using equation numbers 4, 5, and 6 respectively. The variable w_1, w_2, w_3 denotes the synaptic weights, b_3 denotes the bias value at the summing point. The variate $H_1 H_2$ H_3 measures the summing function of the neurons at the hidden layer of the ANN model and X_1, X_2 denotes the input patterns of the neural network.

Fig. 2 exhibits that the mathematical expression outcomes of the perceptron go to the activation function$f(AFs)$. Sum function provides the sum of the product of the edge weight and input parameters (tasks identity and virtual machines identity). The probabilistic outcomes of the resource provisioning in a scalable cloud aura are controlled using the Leaky Relue activation function. Eq. 3, gives the output results. The proposed model contains only one hidden layer, which is feed-forwarded by the previous layer output. The clear picture of the proposed model is shown in Fig. 1. The values of the perceptron measures using the Leaky Relue activation function.

$$H_1 = f(w_a{*}x_1 + w_x{*}x_2 + b_1) = .01{*}(w_a{*}x_1 + w_x * x_2 + b_1) \tag{4}$$

$$H_2 = f(w_b{*}x_1 + w_y{*}x_2 + b_2) = .01{*}(w_b{*}x_1 + w_y{*}x_2 + b_2) \tag{5}$$

$$H_3 = f(w_c{*}x_1 + w_z{*}x_2 + b_3) = .01{*}(w_c{*}x_1 + w_z{*}x_2 + b_3) \tag{6}$$

$$f(H_1), f(H_2), f(H_3), f(O_1) \tag{7}$$

Where w_a, w_b, w_c, w_x, w_y, w_z denote the synaptic weights, b_1, b_2, b_3 denotes the bias value at perceptron 1, 2, 3 as shown in Fig. 1. The output H_1, H_2, H_3 denote the sum function measure at neuron. The sum function with bias value acts as an input to the activation function which provides an output at the perceptron node. Eq. 11 presents the activation function value of the perceptron at the Hidden layer and the next layer to the hidden layer. The computation process inside node performs using Eq. 11. Eq. 11, presents the Leaky Relue activation function. The bio-inspired genetic algorithm offers benefits over artificial neural networks. The weight values are initialized in the range (-1, 2). The variable $b_1$represents the bias value of the perceptron in layer one. The optimization criteria improve using the perceptron based model. The values of the perceptron depend on the activation function (Leaky Relue).

$$\begin{aligned} Y_1 = O_1 &= f(w_1 * H_1 + w_2 * H_2 + w_3 * H_3 + b_3) \\ &= \{.01 * (w_1 * H_1 + w_2 * H_2 + w_3 * H_3 + b_3) \\ &\text{ if } x < 0\} \end{aligned} \tag{8}$$

$$\begin{aligned} Y_1 = O_1 &= f(w_1 * H_1 + w_2 * H_2 + w_3 * H_3 + b_3) \\ &= \{w_1 * H_1 + w_2 * H_2 + w_3 * H_3 + b_3 \text{ if } x \geq 0\} \end{aligned} \tag{9}$$

Eqs. 2–7 provide the results which are optimally fit for the ANN model.

$$Y_1 = A{*}x_1 + B{*}x_2 + C \tag{10}$$

Where Y_1 denotes the output at the output layer, and H_1, H_2, H_3 denote the sum function values at the hidden layer, w1, w2, w3 denote the synaptic weights respectively.

$$Where\ \ A = .01\ \{w_a * w_1 + w_b * w_2 + w_c * w_3\}$$

$$B = .01\ \{w_1 * w_x + w_2 * w_y + w_3 * w_c\},$$

$C = .01\ \{w_1{*}b_1 + w_2{*}b_2 + w_3{*}b_3\}$

The Eq. 10, presents the multi-dimensional straight line. The set of observations fit the straight line on different planes. Hence tasks provisioning in scalable cloud fit the curve shown in Eq. 10. ANN-based model weight of the edges among the perceptron using range $(-1 \le w_{kj} \le 2)$. Where $\forall w_{kj}$presents the connection strength of the perceptron k for j^{th} input. Eqs. 4–7 include the weight function, activation function parameters, and bias parameters. The complexity of the neural network depends on a number of layers and the number of perceptron or neurons in each layer. The proposed model includes three layers (Input layer, Hidden layer, and output layer. The learning rate defines the success of the artificial neural network using bio-inspired techniques. The artificial neural network learns faster when the learning rate increases. The performance study model set the learning rate 0.2 for the optimal global solution. The ANN model is used on a population size of a hundred.

4 Model training

In this phase, the designed neural model is trained with 80 % of the training data prepared to design a trained model. The phase is important because it is responsible for setting the weights of the neural network which defines the accuracy of the neural network. The training process initializes the weight and bias parameters. The process recurrent until the target values are attained. The target values of the proposed model include the instance identity, which takes minimum makespan for the completion of the tasks. The training process predicts an appropriate value of the unknown variables. The training mechanism includes the error correction based mechanism and memory-based mechanism. The optimal trained neural network provides the optimal global solution having the accuracy, stability, and correctness. The available data should be categorized by applying a thumb rule, i.e., 80 % of available data uses for training, and the rest of the 20 % data uses for validation of output.

The neural network training includes the following steps.

1 Give an initial range for the strength of the connection (edge weights $=w_{kj}$) and biases if designing through the use of programming abstraction.
2 Give the required data sets and train the neural network model.
3 Once training of a network is completed, its validation must be performed using the rest of the data set. This is an essential step for efficiency measurement of the network. If desired results are not obtained, the network must be redesigned and trained again and validate the redesigned neural network. So it is a trial and error process. The edge weight and bias parameters adjust if there is a difference between target output and the expected output.
4 Once the Neural based model is validated, and the range of errors is established. The network runs to predict the values. Hence increasing the nodes in a layer and increasing the layers allows the network to cope up with the complex scheduling problems. It also increases the computational space required. Eq. 11, uses for the measurement of the parameters associated with the perceptron node. The activation function of the nodes in each layer measures using the Leaky Relue Activation function. The mathematical expression is shown in Eq. 11.

$$\textit{Leaky}\ \text{Relue}\textit{Activation}\ \text{function} = f(x) = \{\ .01 * x\ \textit{if}\ x < 0,\ x\ \textit{if}\ x \ge 0\ \} \tag{11}$$

Where$f(x)$ represents the activation function applied at the perceptron level and variate $x$denotes the input value at each perceptron. The quality of service measures using optimization criteria makespan and cost pay for resource utilization. The input layer forwards the weight to the hidden layer, and hidden layers forward the results of the perceptron on the output layer. The activation function of a host defines the output of the host which is given as a set of inputs. This prominent feature of the activation function plays a prominent role in computational artificial neural networks. Eq. 11 exhibits the probabilistic activation function which includes the Simple **Relue** and **Leaky Relue** function for the calculation of the perceptron values. The Neural network-based model trains using the requests vector which is the subset of user-defined data sets. The proposed ANN-based resource provisioning model includes two inputs, one output, and one hidden layer. The model is trained for the data set or workload size, one-tenth of the user-defined workload. Two matrices are used for training input and training output. The training input size = 100*2 and training output size includes 100*1. The workload size for the training data set includes 100 tasks, which are 10 percent of the user-defined workload (Fig. 2).

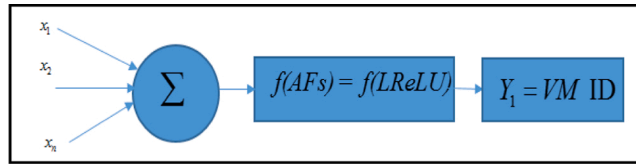


Fig. 2. Mathematical computation at Each Neuron of the ANN model.

5 Error Backpropagation and correction

Here our objective is to obtain the desired output. The desired output is obtained only after a proper neural network is designed. The feed-forward method of the computation process is used in the forward direction. The computation in each of the nodes in the hidden and output layers is performed using the activation function. Fig. 1.1, exhibits the computation function (activation function). The neurons of the adjacent layers are connected using connection weights and biases.

The proposed model follows the error-correction learning mechanism. The error correction learning mechanism improves the quality of service using cost function (δ_k). The goal is to minimize the cost function using the signal flow graph of the ANN model as shown in Fig. 1. Error correction learning mechanism measures the error using Eq. 12.

$$e_k(n) = d_k(n) - y_k(n) \tag{12}$$

$$\delta_k = \frac{1}{2}{*}e_k^2(n) \tag{13}$$

In Eq. 12, and Eq. 13, the parameter $d_k(n)$ denotes the desired output, $y_k(n)$ denotes the calculated output, and $e_k(n)$ is the error, and δ_k is the cost function used in the learning process of the ANN model respectively.

6 Task scheduling

In this phase, the neural network trained model is used to schedule the tasks in real-time

Algorithm of the Proposed Resource Provisioning Technique Neural-GA

Fig. 3 presents the flow diagram of the proposed model. The flow diagram exhibits the pictorial representation of the proposed algorithm GA-ANN which is shown in subsection 4.1. It shows a clear picture of the role of the Genetic algorithm and optimization of the edge weight with the layer perceptron model. Hence the input data is generated using a genetic algorithm. Input data sets of virtual machine id and cloudlet id are provided using the nature-inspired evolutionary technique.

5. Results and discussions

In this section, the performance evaluation of the proposed model is performed. Many criteria are used for task scheduling evaluation. Many

Input: List of Tasks and List of virtual machines.

START {

Generate the initial population

START {

Apply three layer perceptron model

Evaluating GA using fitness function computations

While (Optimization criteria satisfied)}

END

START

Selection

Crossover

Mutation

Compute fitness using the expression.

$distance = cloudletLength / MIPS$ // distance parameter measures the fitness values of the schedules

OUTPUT: Next generation

Repeat the perceptron model for Next generation

END

While (Convergence Condition)

}

END

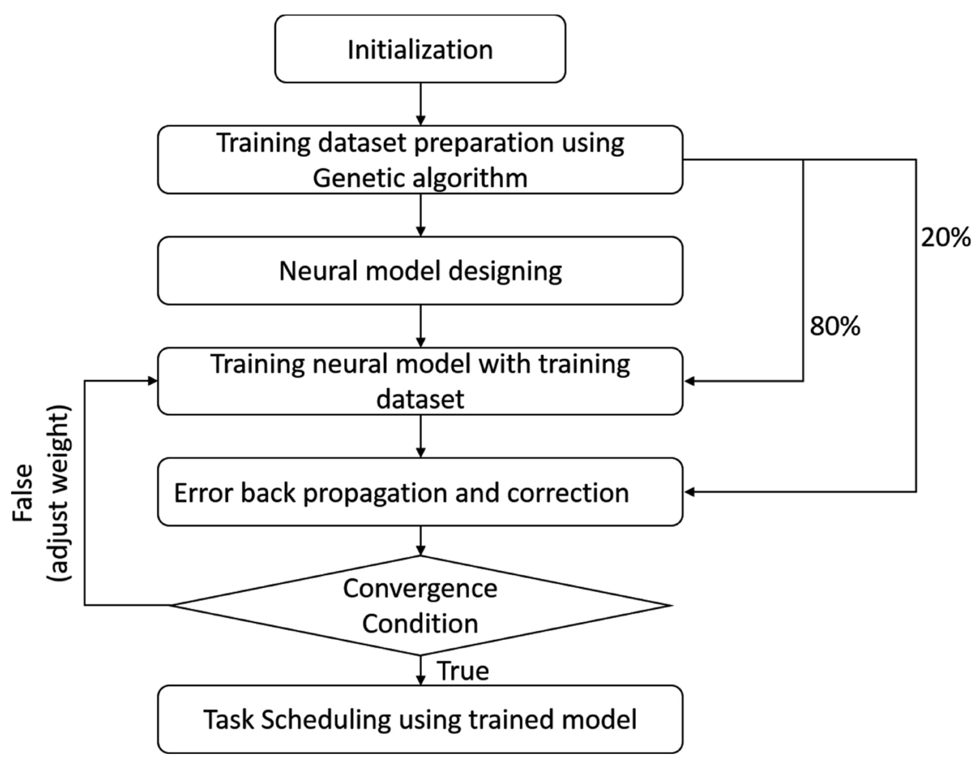


Fig. 3. Flow Diagram of Proposed GA-ANN Scheduling Technique in Cloud.

Table 3
Configuration Parameters of Datacenter.

Datacenter ID	Memory (GB)	RAM (Gb)	Processor (PE)	Hosts	CORE
D1	100000	64	6	2	4
D2	100000	64	6	2	4
D3	100000	64	6	2	4
D4	100000	64	6	2	4
D5	100000	64	6	2	4

Table 4
Virtual Machine Configuration.

Virtual Machine Type	Storage	MIPS per processor	RAM	PE
VM 1	1000	500	1024	1
VM 2	1000	400	512	1
VM 3	1000	600	2048	1

Table 5
Simulation Parameters.

Layers	Input layer:1, Hidden layer:1 Output layer:1
Learning Rate	0.2
Mutation Rate	0.15
Population Size	100
Evolution	100
Neurons Input Layer	2
Neurons Output Layer	50/100
Neurons Hidden Layer	1
Activation Function	$\textit{Leaky}\text{Relue}\textit{Activation}\text{function} = \text{f}(x) = \{\alpha^* x\text{if}x \leq 0, x\text{if}x > 0\} \forall\alpha = 0.01$

of the Metrics are given below:

- Total Execution Time/ Makespan (Millisecond)
- Average Start Time (Millisecond)
- Average Finish Time (Millisecond)
- Scheduling Time (Millisecond)
- Power Consumption (Kwh)

5.1. Simulation setup and performance evaluation

Simulation is performed using Cloudsim 3.0, integrated with NetBeans IDE 8.2. All the experiments are performed using fabricated datasets and real datasets from Parallel Workloads Archive [31] an open source workload log of various datacenter machines around the world. The simulation is conducted with varying load over the cloud with an increasing number of tasks and varying the resource in the cloud with an increasing number of virtual machines. Table 3 shows the basic data center configuration, virtual machine configuration, and a variety of tasks. Table 4 shows the configuration of virtual machines taken into consideration. Table 5 exhibits the details about the ANN model and genetic algorithms like learning rate, mutation rate, population size, number of evolution, number of layers, and count of neuron in each layer.

5.2. Results using fabricated data sets

In this section, the simulation is computed using fabricated tasks and task count. The simulation considered 3 types of task small, medium, and large tasks with task size 1000, 2000, 3000. Where task size refers to the number of instructions in a task. The comparison is done with nature-inspired approaches [22]. Fig. 4 (a) shows a comparison makespan of four different algorithms and performance is measure in ten different scenarios with an increasing number of tasks. Table 6 shows the details about simulation scenarios with a variation of several tasks and a constant number of evolution, population size, virtual machines, and mutation rate. The proposed model provides an optimal solution as shown in Fig. 4 (a) and Fig. 4 (b) proposed model takes the least total execution time/makespan with an increasing number of tasks taking 4 VM's and 8 VM's into count. Performance variation improves while increasing the number of tasks or user requests.

Fig. 4(a) and 4 (b) also exhibit that the results of the model outperform the BB-BC-Cost, Genetic-Cost model, and Genetic-Exe model. Fig. 4 (a) and 4 (b) shows the performance study with a scaling number of tasks and the number of virtual machines in term of resources.

Fig. 5, exhibits the variation of average finish time with population

Table 6
Simulation scenarios using variable Number of Tasks.

Scenario	Population Size	Evolutions	Mutation Rate	Number of Tasks	Number of Virtual Machine
1	100	100	0.15	1000	4
2	100	100	0.15	2000	4
3	100	100	0.15	3000	4
4	100	100	0.15	4000	4
5	100	100	0.15	5000	4
6	100	100	0.15	6000	4
7	100	100	0.15	7000	4
8	100	100	0.15	8000	4
9	100	100	0.15	9000	4
10	100	100	0.15	10000	4

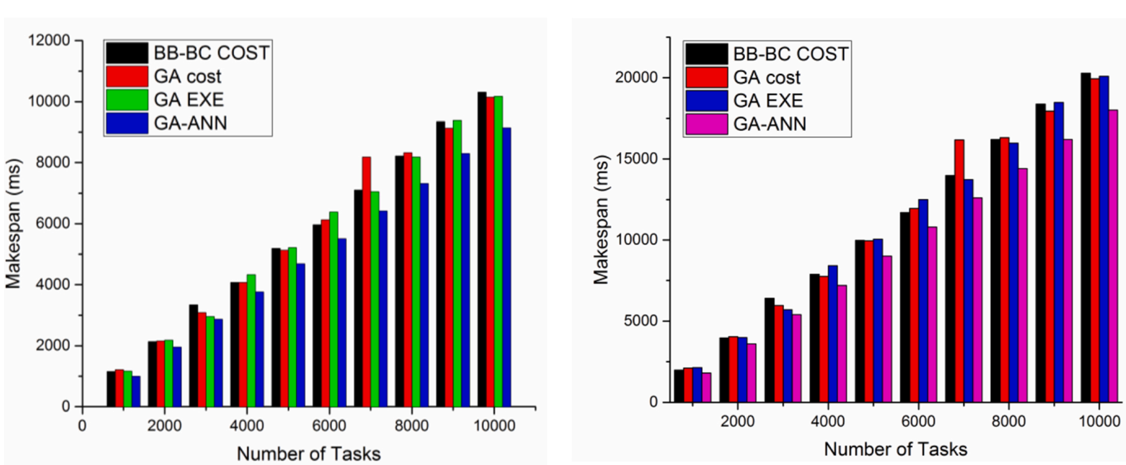


Fig. 4. (a) Makespan (ms) vs number of tasks with 4 VM's. Fig. 4(b) Makespan (ms) vs number of tasks with 8 VM's.

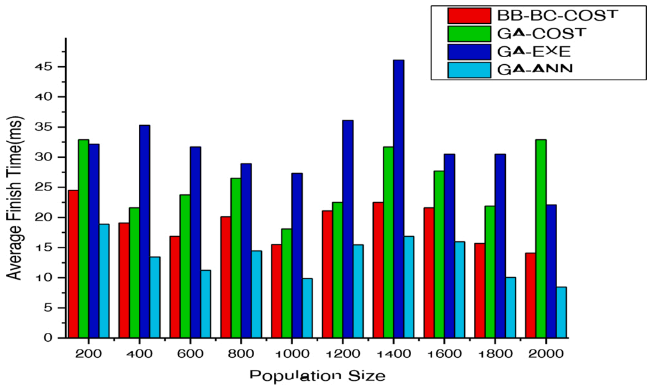


Fig. 5. Average Finish Time (ms) vs population size.

size. Hence the proposed GA-ANN model outperforms the simple nature-inspired meta-heuristic techniques using varying population size.

Table 7 exhibits the simulation results of the presented resource provisioning technique. Simulation results are shown with a variation of average start time with user requests vary from 1000 to 10000. In all the scenario mutation rate is 0.15.

Fig. 6(a) and (b) illustrate the variations of the average start time on increasing the number of tasks with 4 VM's and 8 VM's correspondingly. The neural network-based model provides optimal results with task scaling and resource scaling.

Fig. 7(a) and (b) reveal the simulation results of the presented resource provisioning techniques for the average finish time with varying VM count. Fig. 7 compares the variations of the average finish time on increasing the user requests (tasks) with 4 VM's and 8 VM's correspondingly. The tasks or user requests are mapped from different geographical areas across the globe (Table 8).

5.3. *Results using real workload file (SAN -DIEGO SUPERCOMPUTER CENTER (SDSC) BLUE HORIZON LOGS])*

The performance of the proposed GA-ANN and existing tasks provisioning techniques measures using real workload data sets. In this work, the performance is measured using data sets from a system having 144-node IBM SP, with eight processors per node. The log duration is April 2000 to Jan 2003. The number of jobs included in the log file with datacenter configuration and VM configuration. Table 9 shows the simulation results using ten different scenarios. The number of tasks varies from one thousand to ten thousand. Table 9 exhibits the start time (ms) variations of the proposed neural Genetic approach, BB-BC cost, GA-Cost, and GA-Exe, respectively. Requests generate using Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs) [31].

Fig. 8(a) and (b) shows the variations of average start time with an increasing number of user requests with 4 VM's and 8 VM's correspondingly. The results are obtained using real workload logs. The performance is measured using the average start time of the tasks. Four nature-inspired techniques are used for performance measurement. The proposed GA-ANN outperforms the existing bio-inspired approaches with the least average start time.

Table 10 describes the variations of the average finish time with various user request scenarios. The user requests vary from one thousand to ten thousand. The performance metric finish time is used for the comparative study of the resource provisioning techniques. The resource provisioning technique Neural-GA outperforms using the performance metric average finish time(ms).

Fig. 9(a) and (b) illustrate the variations of the average finish time with the number of submitted tasks with 4 VM's and 8 VM's

Table 7
Simulation Results of the Performance metric (Average Start Time).

Number of Tasks	4 VM's			8 VM's		
	BB-BC-Cost	GA-Cost	GA_ANN	BB-BC-Cost	GA-Cost	GA_ANN
1000	615.92785	625.31514	308.15014	377.96393	336.65757	225.07507
2000	1222.344645	1245.48507	608.150715	624.17232	707.74253	357.07536
3000	1839.252523	1862.08338	906.5555233	960.62626	986.04169	473.27776
4000	2494.214677	2534.358267	1260.109035	1347.10734	1353.17913	668.05452
5000	3046.525542	3124.073372	1531.3126	1602.26277	1601.03669	865.6563
6000	3740.3935	3781.687287	1870.463572	1950.19675	1905.84364	965.23179
7000	4390.243091	4430.214591	2200.979591	2251.12155	2248.1073	1168.4898
8000	4915.602446	4956.012983	2408.497983	2491.80122	2494.00649	1220.24899
9000	5609.18846	5649.914063	2783.913063	2827.59423	2842.95703	1417.95653
10000	6242.641614	6282.838757	3098.453757	3178.32081	3196.41938	1576.22688

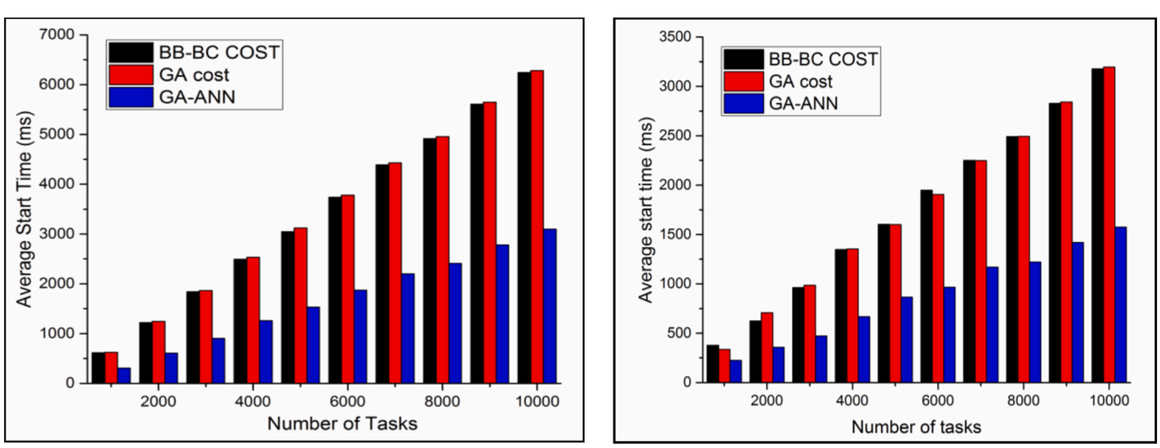


Fig. 6. (a) Average start time (ms) vs number of tasks with 4 VM's. Fig. 6 (b) Average start time (ms) vs number of tasks with 8 VM's.

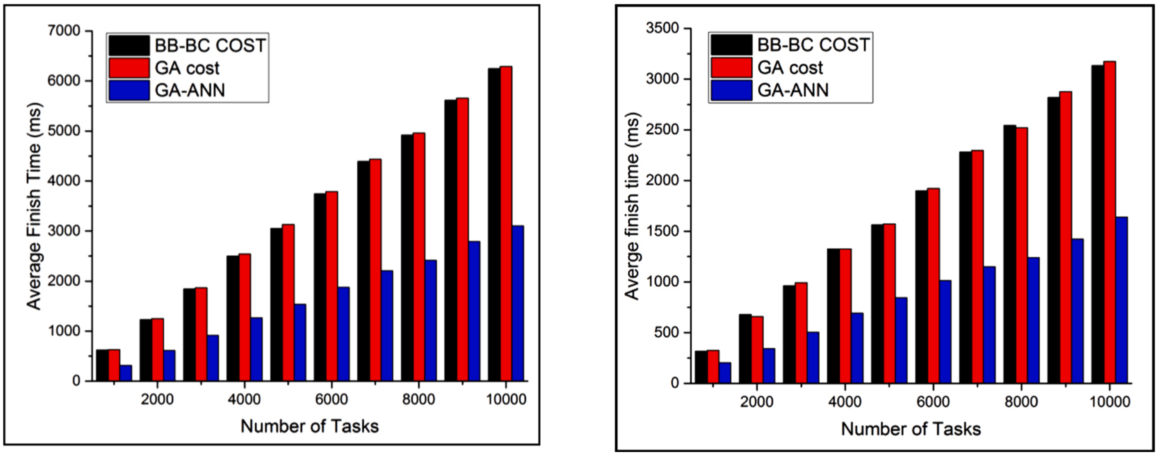


Fig. 7. (a) Average finish time (ms) vs number of tasks with 4 VM's. Fig. 7 (b) Average finish time (ms) vs number of tasks with 8 VM's.

Table 8
Average Finish Time comparison using existing approaches.

Number of Tasks	4 VM's			8 VM's		
	BB-BC-Cost	GA-Cost	GA_ANN	BB-BC-Cost	GA-Cost	GA_ANN
1000	622.15785	630.304	313.49514	316.07893	324.152	203.74757
2000	1228.44465	1250.46372	613.49572	678.22232	659.23186	341.74786
3000	1844.25252	1867.05252	911.90052	961.12626	991.52626	503.95026
4000	2499.1925	2539.37403	1265.45404	1326.59625	1325.68702	692.72702
5000	3051.4448	3129.0576	1536.6576	1563.7224	1571.5288	845.3288
6000	3745.37091	3786.68857	1875.80857	1897.68545	1921.34429	1012.90429
7000	4395.23459	4435.68459	2206.32459	2280.6173	2295.8423	1150.1623
8000	4920.54298	4961.68298	2413.84298	2543.27149	2520.84149	1240.92149
9000	5614.16406	5655.61406	2789.25806	2818.08203	2875.80703	1423.62903
10000	6247.62076	6288.59876	3103.79876	3133.81038	3175.29938	1637.89938

Table 9
Average Start Time (ms) comparison with existing approaches.

Number of Tasks	4 VM's				8 VM's			
	BB-BC COST	GA-Cost	GA-EXE	Neural-GA	BB-BC COST	GA-Cost	GA-EXE	Neural-GA
1000	220.419	323.215	319.1514	431.24597	188.2095	202.6075	213.5757	303.623
2000	820.19147	683.67	776.928	1133.5001	437.09573	368.835	458.464	657.7501
3000	1545.4572	1599.09	1281.058	1433.7845	839.7286	876.545	675.529	814.8923
4000	2316.4746	2206.47	2497.638	1842.6921	1222.2373	1181.235	1279.819	960.3461
5000	2972.3955	2798.309	2661.066	2165.8489	1577.19775	1470.1545	1404.533	1140.924
6000	3017.6094	3575.309	3411.796	2568.379	1525.8047	1837.6545	1732.898	1295.19
7000	4579.7754	5477.9272	4644.11	3515.734	2354.8877	2787.9636	2355.055	1832.867
8000	6051.34	6119.3496	6781.62	4329.688	3076.67	3081.6748	3434.81	2212.844
9000	6875.573	8047.739	8675.449	4501.567	3507.7865	4087.8695	4411.725	2332.784
10000	8497.62	9352.41	7808.518	5862.6685	4295.81	4712.205	3952.259	3010.334

correspondingly. The tasks are submitted on virtual machines using proposed GA-ANN, BB-BC cost, and two other variants of the Genetic approach. The performance metric average finish time acts as a performance measurement parameter. The average finish time is measured using various user requests, which varies from 1000 to 10,000. Table 11 exhibits the performance metric execution time (ms). The tasks are submitted on four virtual machines. The execution time is measured using tasks, which are generated using a real workload data sets logs. The number of requests varies from 1000 to 10,000. The proposed Neural-GA outperforms the existing bio-inspired meta-heuristic techniques.

Fig. 10(a) and (b) illustrate the comparisons of proposed GA-ANN with variations of the number of tasks taking execution time (makespan) as performance metrics with VM count as 4 and 8. The performance metric total execution time improves with increasing the number of user requests. The proposed GA-ANN model outperforms in ten different scenarios as shown in Table 11. The results proclaim that the ANN-based model enhances the performance of the bio-inspired Genetic technique. This sub-section describes the performance metric scheduling time of the tasks on virtual machines. Table 12, exhibits the variations of tasks scheduling time on four virtual machines. The scheduling time varies with an increasing number of user requests. The tasks are assigned on four virtual machines employing the proposed Neural-GA model. The proposed Neural-GA model is compared against three nature-inspired approaches. The results exhibit that the proposed Neural-GA outperforms bio-inspired techniques. The schedule selection criteria (fitness function) measures using simulation time in a scalable IaaS and PaaS model.

Fig. 11 exhibits the variations of the scheduling time in a scalable cloud environment. The performance measures using four scenarios. The proposed GA-ANN provides optimal scheduling time (ms) in a scalable simulation environment.

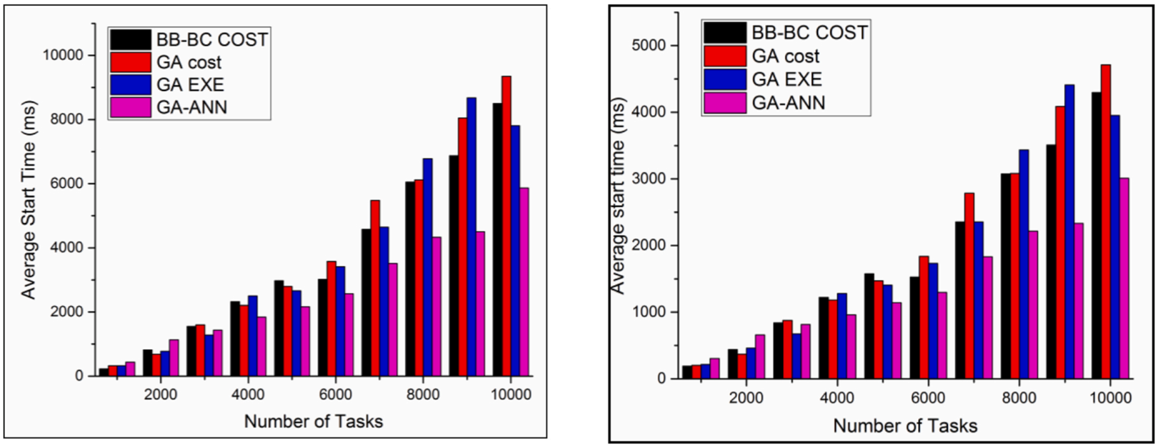


Fig. 8. (a) Average start time (ms) vs number of tasks with 4 VM's. Fig. 8 (b) Average start time (ms) vs number of tasks with 8 VM's.

Table 10
Average Finish Time of Tasks.

Number of Tasks	4 VM's				8 VM's			
	BB-BC COST	GA-Cost	GA-EXE	Neural-GA	BB-BC COST	GA-Cost	GA-EXE	Neural-GA
1000	222.6486	325.64	321.5473	433.07648	184.3243	249.82	187.77365	292.53824
2000	823.2227	686.69	780.206	1135.8813	455.61135	412.345	405.103	603.94065
3000	1549.6016	1603.5479	1284.77	1436.903	859.8008	812.77395	736.385	773.4515
4000	2320.8784	2210.699	2502.692	1847.4879	1168.4392	1140.3495	1330.346	934.74395
5000	2976.816	2802.801	2665.355	2170.7217	1557.408	1430.4005	1426.6775	1138.3609
6000	3021.753	3579.85	3416.628	2572.4106	1553.8765	1889.925	1777.314	1305.2053
7000	4584.981	5483.9326	4649.42	3520.8635	2294.4905	2788.9663	2421.71	1860.4318
8000	6057.064	6125.2915	6788.3115	4335.4053	3111.532	3111.64575	3457.1558	2243.7027
9000	6881.903	8055.0225	8682.877	4508.635	3466.9515	4039.51125	4380.4385	2283.3175
10000	8504.654	9359.78	7814.897	5869.355	4308.327	4777.89	3945.4485	3002.6775

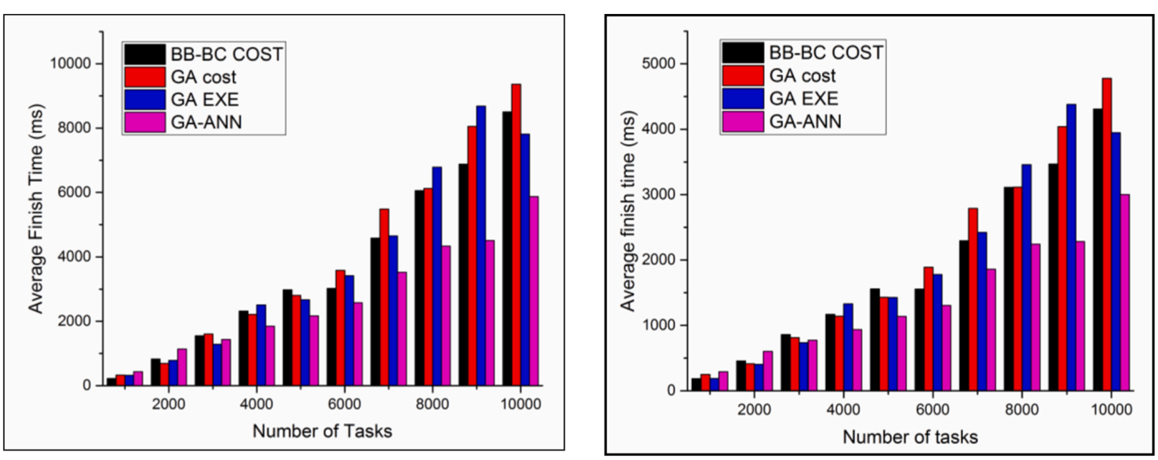


Fig. 9. (a) Average finish time (ms) vs number of tasks with 4 VM's. Fig. 9 (b) Average finish time (ms) vs number of tasks with 8 VM's.

Table 13, exhibits the power consumption by the selected servers having utilization percentage from 0 to 100 %. The results shown in Table 13, are represented graphically in Fig. 12(a) and (b) with 4 VM's and 8 VM's correspondingly. Simulation results are obtained using real workload data sets. Results are shown using two scenarios HP ProLiant G4, and HP ProLiant G5.

The Four nature-inspired meta-heuristic techniques are implemented using Cloudsim 3.0. The proposed Neural-GA exhibits the optimal results using scheduling time as a performance metric. The scheduling time is improved with the number of tasks or user requests generated using Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs). Fig. 10 shows a comparison of power consumed by an existing algorithm and proposed Bio-inspired algorithms. The proposed Neural-GA outperforms the time aware and cost-aware bio-inspired Genetic technique. The performance of the Genetic approach is improved by using the neural network, which is embedded inside the bio-inspired approach.

There is some limitation of the proposed GA-ANN technique i.e. it totally depends on the variety of training data sets at the input layer of the designed neural network for achieving good results. There is no specific rule of determining the configuration of neural network model. The appropriate structure is achieved using training, and error propagation. This limitation can be removed by preparing the data set from dataset of most variety of tasks and VM configurations which directly

Table 11
Total Execution Time (Makespan) of Tasks.

Number of Tasks	4 VM's				8 VM's			
	BB-BC COST	GA-Cost	GA-EXE	Neural-GA	BB-BC COST	GA-Cost	GA-EXE	Neural-GA
1000	253.60475	377.80375	242.78785	370.81149	227.2095	357.6075	263.5757	371.62299
2000	396.54787	402.9175	448.732	565.87503	571.09573	487.835	533.464	737.75005
3000	584.3643	587.2725	572.2645	579.94613	872.7286	944.545	828.529	843.89225
4000	780.11865	747.6175	824.9095	687.17303	1296.2373	1213.235	1443.819	1084.3461
5000	943.59888	940.57725	925.7665	754.96222	1681.19775	1599.1545	1485.533	1223.9245
6000	935.90235	1082.32725	1068.949	868.59475	1671.8047	1908.6545	1829.898	1427.1895
7000	1327.44385	1614.9818	1350.5275	1101.9335	2408.8877	2871.9636	2437.055	1913.867
8000	1698.835	1749.3374	1949.405	1337.922	3165.67	3184.6748	3546.81	2281.844
9000	1928.39325	2309.43475	2367.86225	1387.39175	3600.7865	4218.8695	4481.7245	2450.7835
10000	2376.905	2552.6025	2230.6295	1624.66712	4381.81	4799.205	4091.259	3045.3343

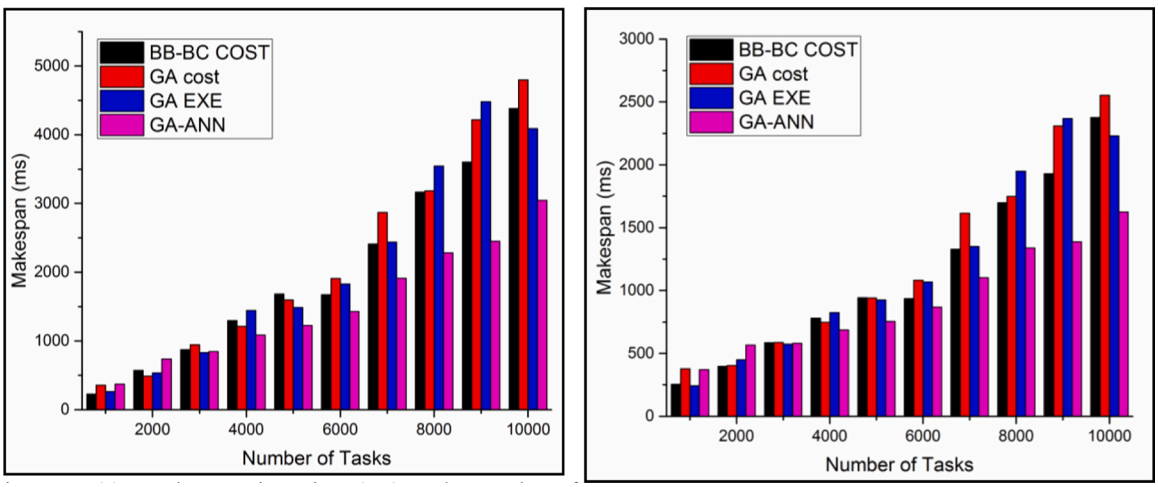


Fig. 10. (a) Total Execution Time (ms) vs the number of tasks with 4 VM's. Fig. 10 (a) Total Execution Time (ms) vs the number of tasks with 8 VM's.

Table 12
Variations of scheduling time versus number of tasks (Generated using standard workload file).

Scenario	Number of Tasks	BB-BC COST Scheduling Time (ms)	GA-Cost Scheduling Time (ms)	GA-EXE Scheduling Time (ms)	Neural-GA Scheduling Time (ms)
1	1000	18439	24612	23684	5625
2	2000	57925	111047	103828	7782
3	3000	134616	268020	263608	13063
4	4000	256342	520898	489158	24159
5	5000	406586	822885	798193	41645
6	6000	594990	1269276	1157798	59300
7	7000	826594	1731555	1639125	84301
8	8000	1115300	2173470	2241534	113836
9	9000	1476236	2830440	2728608	142321
10	10000	1761754	3529092	3513275	183891

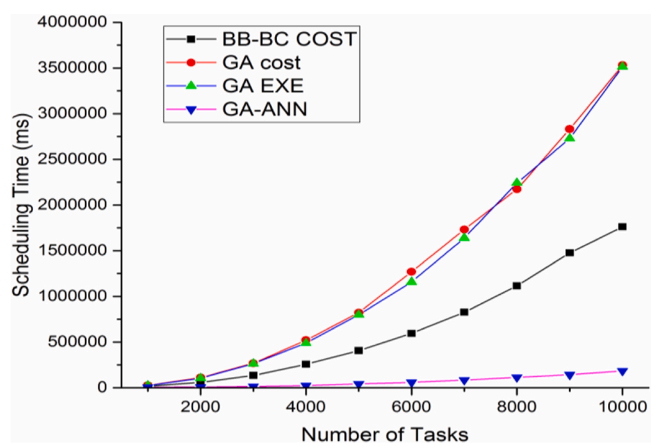


Fig. 11. Number of Tasks (Number of Requests using Workload logs) versus scheduling Time (ms).

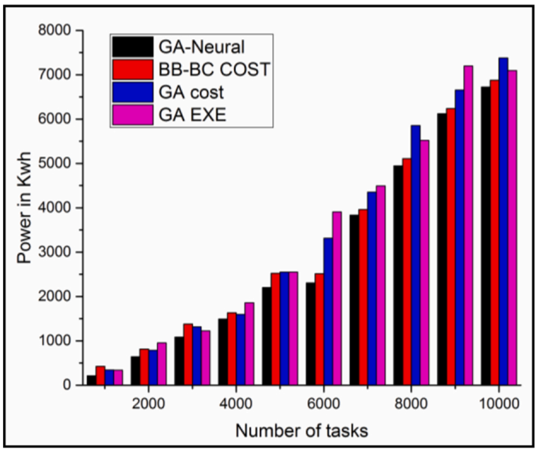
reflects the pattern of upcoming tasks.

6. Conclusion & future works

Scheduling of the tasks on a virtual machine is one of the leading research domain in a scalable cloud environment. Task scheduling is considered a major concern. The task scheduler policy provides an optimal global solution in a service-oriented computing environment. In this work, a review of the literature on task scheduling for scalable cloud infrastructure is done to identify the problem in existing work. In the literature review work, researchers solved the concern of the task provisioning problem using different optimization criteria. Most of the approaches are nature-inspired techniques. It includes a genetic algorithm and ant colony optimization techniques. However, still, there is an opportunity to improve the performance of the bio-inspired meta-heuristic techniques. To cover the problem in existing work a neural network-based approach is proposed. The proposed GA-ANN technique outperforms the Genetic approach and Big-Bang Big-Crunch cost-aware techniques. The training, validation, and prediction is performed using real-time data sets from standard workload file and fabricated data sets at IaaS, PaaS, and SaaS level. The results and discussions section exhibits that the Neural-GA technique outperforms the existing nature-inspired techniques (Genetic, Big-Bang Big-Crunch) in a scalable cloud.

Table 13
Power consumption by the selected servers at different utilization in Watts.

Server	Utilization										
	0 %	10 %	20 %	30 %	40 %	50 %	60 %	70 %	80 %	90 %	100 %
HP ProLiant G4	86	89.4	92.6	96	99.5	102	106	108	112	114	117
HP ProLiant G5	93.7	97	101	105	110	116	121	125	129	133	135



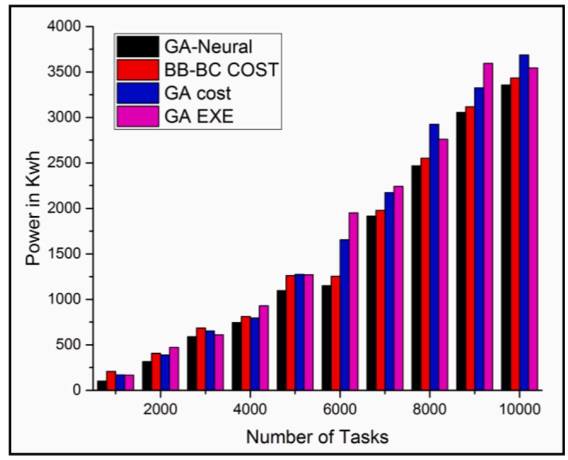


Fig. 12. (a). Number of Tasks vs Power consumed in kilowatt-hours (Kwh) with 4 VM's. Fig. 12 (b). Number of Tasks vs Power consumed in kilowatt-hours (Kwh) with 8 VM's.

Performance is measured using performance metrics average start time, average finish time, and total execution time, and simulation time (ms), and power consumption (Kwh) while increasing the user requests. The results using performance metrics proclaim that the proposed system improves the performance of cloud applications in the real-world with the execution of a task in the least time(ms) and the least power consumption (Kwh). The performance metric execution time improves 30 % using GA-Neural. Hence the proposed model gives an optimal result than the BB-BC cost-aware technique. In future work, the GA-Neural task scheduler policy will be tested in the real cloud computing environment. In this work, our primary concern includes scheduling of the tasks on a virtual machine using neuro bio-inspired GA-ANN. In future work, the proposed algorithm can be used for resource allocation and migration algorithms for better utilization of resources and improve the cost of running cloud applications with the least resources.

Author statement

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication.

The revised copy is updated keeping in mind and answering all the reviewer comments. All authors have equally contributed on their part to update the article. The article is updated with all the required modification with valuable comments from reviewers.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

Appendix A. Supplementary data

Supplementary material related to this article can be found, in the online version, at doi:https://doi.org/10.1016/j.suscom.2020.100431.

References

[1] K. Stanoevska-Slabeva, T. Wozniak, S. Ristol, Grid and cloud computing: a business perspective on technology and applications, Grid Cloud Comput. A Bus. Perspect. Technol. Appl. (2010) 1–274.

[2] R. Buyya, Introduction to the ieee transactions on cloud computing, IEEE Trans. Cloud Comput. 1 (2013) 3–21.

[3] P.S. Rawat, P. Dimri, G.P. Saroha, Virtual machine allocation to the task using an optimization method in cloud computing environment, Int. J. Inf. Technol. (2018).

[4] Z. Chenhong, Z. Shanshan, L. Qingfeng, X. Jian, H. Jicheng, Independent tasks scheduling based on genetic algorithm in cloud computing, Proc. - 5th Int. Conf. Wirel. Commun. Netw. Mob. Comput. WiCOM 2009 (2009) 0–3.

[5] A. Sathya Sofia, P. GaneshKumar, Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II, J. Netw. Syst. Manag. 26 (2018) 463–485.

[6] J. Praveenchandar, A. Tamilarasi, Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing, J. Ambient Intell. Humaniz. Comput. (2020).

[7] M. Lin, J. Xi, W. Bai, J. Wu, Ant colony algorithm for multi-objective optimization of container-based microservice scheduling in cloud, IEEE Access 7 (2019) 83088–83100.

[8] X.F. Liu, Z.H. Zhan, J.D. Deng, Y. Li, T. Gu, J. Zhang, An energy efficient ant colony system for virtual machine placement in cloud computing, IEEE Trans. Evol. Comput. 22 (2018) 113–128.

[9] L. Hongyou, W. Jiangyong, P. Jian, W. Junfeng, L. Tang, Energy-aware scheduling scheme using workload-aware consolidation technique in cloud data centres, China Commun. 10 (2013) 114–124.

[10] L. Zuo, L. Shu, S. Dong, C. Zhu, T. Hara, Special section on big data services and computational intelligence for industrial systems a multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing, IEEE Access 3 (2015).
[11] T. Kaur, I. Chana, Energy aware scheduling of deadline-constrained tasks in cloud computing, Cluster Comput. 19 (2016) 679–698.
[12] S. Singh, I. Chana, A survey on resource scheduling in cloud computing: issues and challenges, Int. J. Grid Comput. Appl. 14 (2016) 217–264.
[13] A.M. Senthil Kumar, M. Venkatesan, Multi-objective task scheduling using hybrid genetic-ant colony optimization algorithm in cloud environment, Wirel. Pers. Commun. 107 (2019) 1835–1848.
[14] J. Praveenchandar, A. Tamilarasi, Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing, J. Ambient Intell. Humaniz. Comput. (2020).
[15] M.H. Fazel Zarandi, A.A. Sadat Asl, S. Sotudian, O. Castillo, A State of the Art Review of Intelligent Scheduling, Springer, Netherlands, 2020.
[16] Y.S. Jiang, W.M. Chen, Task scheduling for grid computing systems using a genetic algorithm, J. Supercomput. 71 (2015) 1357–1377.
[17] P. Singh, M. Dutta, N. Aggarwal, A review of task scheduling based on meta-heuristics approach in cloud computing, Knowl. Inf. Syst. (2017).
[18] P.M. Rekha, M. Dakshayini, Efficient task allocation approach using genetic algorithm for cloud environment, Cluster Comput. 22 (2019) 1241–1251.
[19] M. Masdari, F. Salehi, M. Jalali, M. Bidaki, A survey of PSO-based scheduling algorithms in cloud computing, J. Netw. Syst. Manag. 25 (2017) 122–158.
[20] X. Sui, D. Liu, L. Li, H. Wang, H. Yang, Virtual machine scheduling strategy based on machine learning algorithms for load balancing, EURASIP J. Wirel. Commun. Netw. (2019).
[21] P. Singh, A load balancing analysis of cloud base application with different service broker policies, 135, 2016, pp. 11–15.
[22] M. Kalra, S. Singh, A review of metaheuristic scheduling techniques in cloud computing, Egypt. Informatics J. 16 (2015) 275–295.
[23] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, ICML Work. Deep Learn. Audio, Speech Lang. Process. 28 (2013).
[24] M. Melnik, D. Nasonov, Workflow scheduling using neural networks and reinforcement learning, Procedia Comput. Sci. 156 (2019) 29–36.
[25] H.F. Sheikh, H. Tan, I. Ahmad, S. Ranka, P. Bv, Energy- and performance-aware scheduling of tasks on parallel and distributed systems, ACM J. Emerg. Technol. Comput. Syst. 8 (2012).
[26] H.F. Sheikh, I.A. Fellow, D. Fan, An Evolutionary Technique for Performance - Energy - Temperature Optimized Scheduling of Parallel Tasks on Multi - Core Processors, 2015, pp. 1–14.
[27] H.F. Sheikh, I. Ahmad, Z. Wang, S. Ranka, An overview and classification of thermal-aware scheduling techniques for multi-core processing systems, Sustain. Comput. Informatics Syst. 2 (2012) 151–169.
[28] https://www.cse.huji.ac.il/labs/parallel/workload/l_sdsc_blue/.
[29] M. Sharma, R. Garg, An artificial neural network based approach for energy efficient task scheduling in cloud data centers, Sustain. Comput. Informatics Syst. 26 (2020), 100373.
[30] L. Mao, Y. Li, G. Peng, X. Xu, W. Lin, A multi-resource task scheduling algorithm for energy-performance trade-offs in green clouds, Sustain. Comput. Informatics Syst. 19 (2018) 233–241.
[31] Parallel Workloads Archive: https://www.cse.huji.ac.il/labs/parallel/workload/.
[32] S.G. Domanal, G.R.M. Reddy, An efficient cost optimized scheduling for spot instances in heterogeneous cloud environment, Future Gener. Comput. Syst. 84 (2018) (2018) 11–21.
[33] P. Zhang, M. Zhou, Dynamic cloud task scheduling based on a two-stage strategy, Ieee Trans. Autom. Sci. Eng. 15 (2) (2017) 772–783.
[34] M. Sharma, R. Garg, An artificial neural network based approach for energy efficient task scheduling in cloud data centers, Sustain. Comput. Informatics Syst. 26 (2020), 100373.
[35] L. Bao, C. Wu, X. Bu, N. Ren, M. Shen, Performance modeling and workflow scheduling of microservice-based applications in clouds, IEEE Trans. Parallel Distrib. Syst. 30 (2019) 2101–2116.
[36] S. Guo, J. Liu, Y. Yang, B. Xiao, Z. Li, Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing, IEEE Trans. Mob. Comput. 18 (2019) 319–333.
[37] P. Kuang, W. Guo, X. Xu, H. Li, W. Tian, R. Buyya, Analyzing energy-efficiency of two scheduling policies in compute-intensive applications on cloud, IEEE Access 6 (2018) 45515–45526.
[38] S. Pang, W. Li, H. He, Z. Shan, X. Wang, An EDA-GA hybrid algorithm for multi-objective task scheduling in cloud computing, IEEE Access 7 (2019) 146379–146389.
[39] T.A.L. Genez, L.F. Bittencourt, N.L.S. Da Fonseca, E.R.M. Madeira, Estimation of the available bandwidth in Inter-Cloud links for task scheduling in hybrid clouds, IEEE Trans. Cloud Comput. 7 (2019) 62–74.
[40] H.R. Faragardi, M.R. Saleh Sedghpour, S. Fazliahmadi, T. Fahringer, N. Rasouli, GRP-HEFT: a budget-constrained resource provisioning scheme for workflow scheduling in IaaS clouds, IEEE Trans. Parallel Distrib. Syst. 31 (2020) 1239–1254.
[41] Q. Wu, M. Zhou, Q. Zhu, Y. Xia, J. Wen, MOELS: Multiobjective Evolutionary List Scheduling for Cloud Workflows, IEEE Trans. Autom. Sci. Eng. 17 (2020) 166–176.
[42] V. Arabnejad, K. Bubendorfer, B. Ng, Budget and deadline aware e-Science workflow scheduling in clouds, IEEE Trans. Parallel Distrib. Syst. 30 (2019) 29–44.
[43] S.K. Mishra, D. Puthal, B. Sahoo, P.P. Jayaraman, S. Jun, A.Y. Zomaya, R. Ranjan, Energy-efficient VM-placement in cloud data center, Sustain. Comput. Informatics Syst. 20 (2018) 48–55.
[44] M.K. Gupta, A. Jain, T. Amgoth, Power and resource-aware virtual machine placement for IaaS cloud, Sustain. Comput. Informatics Syst. 19 (2018) 52–60.
[45] N. Mansouri, M.M. Javidi, Cost-based job scheduling strategy in cloud computing environments, Distrib. Parallel Databases 38 (2020) 365–400.
[46] Y. Lu, N. Sun, An effective task scheduling algorithm based on dynamic energy management and efficient resource utilization in green cloud computing environment, Cluster Comput. 22 (2019) 513–520.
[47] M.R. Shirani, F. Safi-Esfahani, Dynamic Scheduling of Tasks in Cloud Computing Applying Dragonfly Algorithm, Biogeography-based Optimization Algorithm and Mexican Hat Wavelet, Springer US, 2020.