

## Performance evaluation of an adopted model based on big-bang big-crunch and artificial neural network for cloud applications

Pradeep Singh Rawat<sup>1,5</sup>, Robin Singh Bhadoria<sup>2</sup>, Punit Gupta<sup>3,\*</sup>, G. P. Saroha<sup>4</sup>

<sup>1</sup>Dept. of Computer Science & Engineering ,Uttarakhand Technical University, India

<sup>2</sup>Dept. of Computer Science & Engineering, Birla Institute of Applied Sciences, India

<sup>3</sup>Dept. of Computer and Communication Engineering, Manipal University Jaipur, India

<sup>4</sup>Computer Center, Maharishi Dayanand University, Rohtak, Haryana, India

<sup>5</sup>School of Computing, DIT University Dehradun, Uttarakhand, India

\*Corresponding author: punitg07@gmail.com

### Abstract

High-performance computing is changing the way we compute. In the past decade, the cloud computing paradigm has changed the way we compute, communicate, and technology. Cover real-world problems. There are still many complex challenges in the cloud computing paradigm. Improving effective planning strategies is a complex problem in the service-oriented computing paradigm. In this article, our research focuses on improving task scheduler strategies to improve the performance of cloud applications. The proposed model is inspired by an artificial neural network-based system and astrology base scheduler Big-Bang Big-Crunch. The results show that the proposed strategy based on BBBC and neural network is superior to the method based on astrology (BigBang BigCrunch costaware), genetic cost and many other existing methods. The proposed BB-BC-ANN model is validated using standard workload file (San Diego Supercomputer Center (SDSC) Blue Horizon logs). The results show that the proposed BB-BC-ANN model performs better than some of the existing approaches using performance indicators like total completion time (ms), average start time (ms), average finish time(ms), scheduling time(ms), and total execution time(ms).

**Keywords:** Ant Colony Optimization (ACO); Activation Functions (AFs); Artificial Neural Network (ANN); Big-Bang Big-Crunch (BB-BC); leaky ReLU; power consumption; scheduling.

### 1. Introduction

Cloud computing is a scalable and reliable computing platform providing support to the majority of services in the world. It enables users to dynamically allocate and use resources that are remotely located over the globe. Cloud is a distributed environment with a scalable, pay-per-use, and reliable computing environment. A cloud application is involved to provide computing in real-time. Cloud applications require efficient and cost-effective computing environments that provide a better cost-effective and reliable solution to the user. The resource allocation in the cloud computing environment depends on resources used, cost of resources and scheduler policy to improve the performance of the system. Utility

computing allows end-users to use the pay-as-you-go pricing model for infrastructure, platform, and application services. The local or global optimal point provides the solution (Ajeena Beegom and Rajasree, 2014) Ang, Por & Liew, 2017). User requirements may test directly on a real cloud computing environment, for example, Microsoft Azure, Amazon EC2, but it increases the cost overhead. The scalable simulation of the cloud environment reduces the cost. Cloud computing provides dynamic services using virtual resources over the Internet (Ang, Por & Liew, 2017). The cloud service providers provide the facilities for the network, computing, and storage as a service to the end-users. Liu *et al.* presented

information from virtual computing environments and end-user experiences using an ant colony optimization approach (Liu, & Wang (2008, October). Efficient resource provisioning in a scalable environment explores the nature-inspired optimization approaches. Our primary focus is to achieve an objective function using scalable cloud simulation. So before shifting our application to the real cloud, the scheduling policy tests perform on a scalable cloud environment. Zhao et al. demonstrated the bio-inspired strategies, which reduce the over resource provisioning overhead. The author focused on independent task mapping using a bio-inspired genetic approach (Zhao et al., 2009). Dynamic and evolution based meta-heuristic approaches improve the quality of service using distinctive reinforcement learning-based dynamic resource management. The authors primarily focused on resource provisioning in a scalable simulation environment. Singh and Chana depict a broad methodical literature survey of resource management in the area of cloud in general and cloud resource scheduling specifically. It helps the researchers to find the important characteristics of resource scheduling approaches (Zhao et al., 2016). The cloud entity, datacenter broker manages the life cycle of the virtual machine for the scheduling of tasks on the respective virtual machine. It assures the minimization of operational cost and task scheduling time (Kruekaew & Kimpan, 2014). There exist various solutions for resource allocation in a cloud computing environment to improve the performance in the cloud. But most of the algorithms do not take into consideration the previous behavior of incoming requests and how the resources are allocated to achieve better utilization and efficiency. In existing static, dynamic, and meta-heuristic approaches performance metrics time, cost, and network latency are considered which play an important role in remote computing. In this article, we proposed an artificial neural network and big-bang big-crunch based hybrid task scheduling model which fine-tunes the performance metrics used in traditional and existing task scheduling approaches. The proposed model takes into consideration the network delay and time as performance metrics. This work is organized as follows. First, the researchers introduce the service-oriented architecture of scalable cloud infrastructure. Then, we briefly discuss

efficient resource provisioning as a challenging issue. The bio-inspired ANN model plays a prominent role in resource provisioning. Resource provisioning in a scalable cloud environment is explained in the related work section. Subsequently, we provided details about the human brain-based soft computing model with meta-heuristic approaches. The results and discussions section includes the validation of the big-bang big-crunch neural network-based task provisioning on a scalable cloud. Lastly, the conclusion and future works are discussed. The key contributions of our proposed BB-BC-ANN model can be summarized as follows: The proposed model is based on the astrology-based artificial neural network model. The neural computing model takes the cloud tasks and set of virtual machines as an input parameter in a scalable cloud computing environment. Our proposed time-efficient model focuses on performance metrics total completion time (ms), average start Time (ms), average finish time (ms), scheduling time(ms), and total execution time(ms). The objective function also depends on resource utilization cost in the duration of execution of the tasks.

## 2. Background

Cloud computing is a promising paradigm for modern industries to achieve high performance at an affordable cost with a scalable and flexible environment. It is most likely the priority of an organization to reduce the operational cost using the scalable features of the cloud. It provides new opportunities to the developer, end-user, and other stakeholders. This technology is still growing in which computing paradigm encapsulates the application software inside a virtual machine and datacenters. In real-world cloud is a heterogeneous combination of various services. It provides various storage, networking, processing services to the user based on a pay-per-use model. Cloud computing is responsible to execute the user task with high computation and the least cost as promised to the user without compromising the quality matrices. To manage the cloud resources, task scheduling plays an important role to improve the performance of the cloud and the quality of service to users. The dynamic scheduling algorithm provides a better quality of service than the static scheduling techniques (Kalra & Singh, 2015). The prominent benefits of scheduling algorithms are to achieve optimal performance

computing and optimal system throughput. Jin et al. proclaimed a general technique that captures the tradeoff between time and cost (Zhang *et al.*, 2015). Authors have also presented offline and online solutions for specific problems by exploiting special and temporal information. Jin et al. have covered virtual machine assignment and scheduling techniques to reduce the operational cost for improvement of performance-interference in cloud datacenters. The optimization is based on the execution time, and the cost is yet to improve (Yu & Buyya, 2006). Throughput measures using time and operational cost. Cloud computing may support a homogenous, heterogeneous, centralized, distributed, and hierarchical computing environment. Traditional job scheduling algorithm provides efficient scheduling in the cloud. The provisioning techniques are categorized into batch mode heuristic techniques and online mode heuristic techniques. The scheduling algorithms assure the quality of service improvement (Domanaal & Reddy, 2018). The algorithms are specified as first-come-first-service (FCFS), Round Robin scheduling (RR), Min-Min, and Max-Min and batch mode heuristic techniques. The first-come-first-service algorithm maps the job in a queue which comes first is served first. Tasks dispatch in a first in first out manner. The processing unit allocates to the tasks for a time slice or time quantum while using the round-robin approach. The Min-Min algorithm selects the minimum of the minimum, which executes first, and the Max-Min algorithm selects the maximum of the minimum to execute first. The priority scheduling algorithm assigns priority and executes in the same order. The tasks having the same priority executed on a first-come, first-serve mode. Sometimes we prefer the shortest-job-first (SJF) algorithm, which is a special case of the general priority scheduling algorithm. The tasks are mapped on a central processing unit on a priority basis. It depends on the burst time of the tasks. (Gu *et al.*, 2012). exhibited the load balancer policy in a static environment. The quality of service is measured using the average response time. Still, there is a scope for the efficient allocation of the virtual machine using meta-heuristic techniques (Gu *et al.*, 2012). Cost and task execution time are measured using the platform and application parameters. Both the optimization criteria and the selection operators depend on each other. The mapping process is executed using various scenarios. Cloud scenarios include datacenter configuration, resource allocation policy of the tasks on virtual machines. The deployment process includes Internet characteristics for quality of service measurement. It includes bandwidth, delay matrix measurement among the different time zone (Gu *et al.*, 2012). revealed the status of the virtual machines in a cloud computing environment. The authors have also considered the active load balancer as a key focus area of the work. There is a scope to implement the policy which supports the dynamic environment and scalable simulation. Researchers described the load balancing scheme for the private cloud. The performance measurement criteria include utilization standards. The authors primarily focused on bio-inspired meta-heuristic approaches for the performance measurement of the scalable cloud (Kaur *et al.*, 2010). The authors considered nature-inspired ant colony optimization and particle swarm optimization techniques for the quality of service improvement (Lu & Gu, 2011) (Liang *et al.*, 2006). The static load balancing approach does not perform well in a dynamic and distributed environment. Beloglazov *et al.* exhibited multi-objective function minimization, which combines the energy consumption and execution time (Beloglazov *et al.*, 2012). Kalra *et al.* proclaimed a survey on meta-heuristic algorithms. (Kalra & Singh, 2015). (Beheshti & Shamsuddin, 2013) explained the population-based meta-heuristics algorithms for cloud scheduling which includes genetic algorithm, PSO, ACO, and many more (Kalra & Singh, 2015) covered the detailed survey on first come first serve (FCFS), round robin, local search approach, Max-Min, stochastic hill climbing (SHC), and soft computing approach with the genetic algorithm (GA). The author also presented the various optimization criteria for the optimal global solution. (Mosa & Paton, 2016) demonstrated optimized energy and SLA-aware virtual machine scheduling strategy that focuses on system resource distribution. Performance is measured using energy consumption and service level agreement violations (Singh *et al.*, 2016). Arabnejad et al. presented a new heuristic scheduling approach, budget deadline aware scheduling in Infrastructure as a Service (IaaS) clouds (Arabnejad *et al.*, 2019) presented

a comprehensive survey of task scheduling strategies and the associated metrics suitable for cloud computing environments (Arunarani *et al.*, 2019; Buyya, 2013) presented opportunities and challenges in cloud computing. Dabbagh *et al.*, modeled an energy-aware resource provisioning framework for cloud datacenter. The performance is evaluated using real data sets from Google cluster traces (Dabbagh *et al.*, 2015) presented a novel resource provisioning mechanism and a workflow scheduling algorithm. The performance metrics include makespan and budget constraints Faragardi *et al.*, 2020). (Genez *et al.*, 2019) presented a mechanism to cope with imprecise information about the available bandwidth and its impact on makespan and cost. Juarez *et al.* presented a real-time dynamic scheduling system to execute efficiently task-based applications on distributed computing platforms to minimize energy consumption (Juarez *et al.*, 2018). (Mishra *et al.*, 2018) studied the energy consumption in a cloud environment based on varieties of services and achieved the provisions to promote green cloud computing. The proposed technique minimizes the makespan of the cloud system and reduces energy consumption. (Liu, *et al.*, 2018) presented an approach based on ant colony optimization (ACO) to solve the virtual machine placement problem. The presented approach reduces the number of host machine requirements. (Sahni & Vidyarthi, 2018) proposed a dynamic cost-effective deadline-constrained heuristic approach for scheduling a scientific workflow in a public Cloud. (Senthil Kumar & Venkatesan, 2019) presented a utility-based scheduler output and finds the best task allocation method based on response time, total completion time, and throughput. (Shojafar *et al.*, 2015). presented a hybrid approach that is based on a fuzzy theory and a genetic algorithm (GA). The presented approach aims to perform optimal load balancing considering execution time and cost. (Singh *et al.*, 2017). described a review of using meta-heuristics techniques for scheduling tasks in cloud computing. A taxonomy and comparative review are presented on meta-heuristic approaches. Rawat *et al.*, 2020) exhibited the provisioning of tasks on a virtual machine. The Big-Bang Big-Crunch-cost model is proposed for efficient resource allocation. Sreelatha, 2017) presented a task scheduling algorithm called W-Scheduler based on the multi-objective model and the whale

optimization algorithm. The performance is measured using minimum makespan and cost. (Wei *et al.*, 2018) proposed a heterogeneous resource allocation approach for multi-resource allocation. The multi-resource allocation includes virtual machine allocation. (Wu *et al.*, 2020) proposed a multi-objective evolutionary list scheduling approach using performance metrics cost and makespan. (Xiao, Song & Chen, 2013) presented a system that uses virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. (Amiri *et al.*, 2018) developed a model based on neural networks for rainfall prediction with better accuracy. (Sri *et al.*, 2017) has proposed a speculation based task scheduling for computationally intensive processes that may be cloud, grid, or any other distributed system. (Soroush, H. M. (2013). has proposed a stochastic task scheduling mechanism that may be used to optimize time complexity of the system which may be any distributed or parallel computing system. (Zhou, Li, Zhu, Xie, Abawajy & Chowdhury, 2020) proposed an approach that leverages the modified G A a lgorithm c ombined w ith a greedy strategy to optimize the task scheduling process. Performance is measured using total completion time, average response time. This hybrid BB-BC-ANN approach is compared with the cost-aware BB-BC and bio-inspired intelligence systems, and TOPSIS and nature-inspired particle swarm optimization approach. The bio-inspired intelligence systems and the human brain impressive Neural-BB-BC-Cost aware technique improves performance metrics time(ms), cost(\$), and network delay(ms). Our principle goal focuses on time and cost tradeoff in a cloud computing environment using a constant IaaS cloud model.

### 3. Proposed Model

The efficient s cheduling o f t asks o n virtual machines is a challenging concern in a cloud computing environment. The literature survey has covered static, dynamic, and bio-inspired meta-heuristic techniques. There is an opportunity for further improvement in the quality of service. Nature-inspired population-based meta-heuristic techniques provide a local optimal and optimal global solution. Optimization criteria improve

further using a meta-heuristic technique with human brain computation. The human brain's impressive artificial neural network with meta-heuristic approaches provide better results than standalone nature-inspired meta-heuristic techniques. Perceptron and their connection strength (edge weight), and bias values adjust for input data sets prepared using astrology base big-bang big-crunch cost-aware approach. Mutation and fitness selection are performed for the development of the specific schedule. The trained neural network uses for the prediction of the virtual machine, and model the various non-linear applications. At the end of the training or training process, a legal output is created in the output layer. The ANN model using the bionic heuristic method achieved the best results. Researchers are working hard to solve the task scheduling problem in scalable cloud scenarios. Artificial intelligence technologies, such as genetic algorithm, ant colony and BigBang BigCrunch cost accounting, solve task scheduling problems and find the best resource allocation. Our goal is to improve the performance of meta-heuristics based on biological detection with the help of neural networks.

The proposed BB-BC-ANN model is divided into the following phases:

1. Initialization
2. Training dataset preparations
3. Neural model design
4. Neural model Training
5. Error backpropagation and correction
6. Task scheduling

### 3.1 Initialization

In this phase, all the hyper parameters (learning rate, hidden layers, number of the hidden unit, activation function), input layer, and output layer weight, and bias values are initialized.

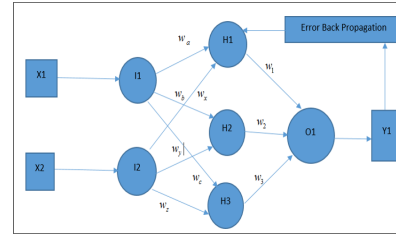
### 3.2 Training dataset preparations)

In this phase, the training data set is prepared using the Big-Bang Big-Crunch cost-aware approach. The fitness function values are measured using network\_cost and task completion\_time parameters as shown in equation 9. The input to the BB-BC cost-aware approach contains a list of tasks and a

list of virtual machines. The output of the BB-BC cost-aware approach is the training data set which is used to train the neural model. As shown in figure 2, 20% of tasks are used for the preparation of training data sets.

### 3.3 Neural model design

Figure 1, reveals the two layers of the artificial neural network. The three layers of the artificial neural network-based model improve the performance of the bio-inspired genetic algorithm and BB-BC Cost-aware model. The performance of ANN using biological methods is measured using the expensive BigBang BigCrunch method, expensive genetics and genetic running time. Artificial neural networks are trained at different learning rates. The accuracy of the output depends on the training parameters. The network has several levels and several levels. Nodes at each level. Performance is affected by the learning mechanism.



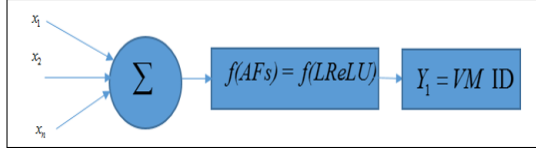
**Fig. 1.** Artificial Neural Network Architecture

The mathematical and computational representation of the proposed BB-BC-ANN model schedules the tasks in a scalable cloud aura. Equation 1 exhibits the output using an activation function.

$$NN(X_1, X_2) = LeakyRelu(w_1 * H_1 + w_2 * H_2 + w_3 * H_3) \quad (1)$$

Where  $H_1$ ,  $H_2$  and  $H_3$  represent the sum function values of neurons in the hidden layer using equations 2, 3, and 4 respectively. As shown in equation 1,  $x_1$ ,  $x_2$  shows the input features i.e. virtual machine id and task identity.  $w_1, w_2$ , and  $w_3$  are the weight parameters of the neural network model and

Figure 2 exhibits that the mathematical expression outcomes of the perceptron go to the activation function  $F(AF)$ . The values of the perceptron



**Fig. 2.** Activation Function at Each Neuron.

measures using the leaky ReLU activation function can be described as follows:-

$$H_1 = f(w_a * x_1 + w_x * x_2 + b_1) \\ = 0.1 * (w_a * x_1 + w_x * x_2 + b_1) \quad (2)$$

$$H_2 = f(w_b * x_1 + w_y * x_2 + b_1) \\ = 0.1 * (w_b * x_1 + w_y * x_2 + b_1) \quad (3)$$

$$H_3 = f(w_c * x_1 + w_z * x_2 + b_1) \\ = 0.1 * (w_c * x_1 + w_z * x_2 + b_1) \quad (4)$$

Where  $F(H_1), F(H_2), F(H_3)$  present the activation function values of the perceptron at the hidden layer and the next layer to the hidden layer. The computation process inside the node is performed using equation 1. Where equation 1 presents the Leaky ReLU activation function. The bio-inspired genetic algorithm offers benefits over artificial neural networks. The weight values are initialized in the range (-1, 2) using error backpropagation. The initial value is -1 but during training and error correction the weight may extend up to 2 for a neuron. The variable  $b$  represents the bias value of the perceptron in layer one. The optimization criteria improve using the hyperparameters learning rate, several hidden layers, number of hidden units, activation function at the hidden unit of the neural network model, and several iterations respectively. The values of the perceptron depend on the activation function (Leaky ReLU).

$$Y_1 = O_1 = f(w_1 * H_1 \\ + w_2 * H_2 + w_3 * H_3 + b_3) = 0.1 * (w_1 * H_1 \\ + w_2 * H_2 + w_3 * H_3 + b_3) \text{ if } x < 0 \quad (5)$$

$$Y_1 = O_1 = f(w_1 * H_1 \\ + w_2 * H_2 + w_3 * H_3 + b_3) = (w_1 * H_1 \\ + w_2 * H_2 + w_3 * H_3 + b_3) \text{ if } x \geq 0 \quad (6)$$

$$Y_1 = A * X_1 + B * X_2 + c \quad (7)$$

Where

$$A = 0.1 * (w_a * w_1 + w_b * w_2 + w_c * w_3), \\ B = 0.1 * (w_a * w_x + w_b * w_y + w_c * w_z), \\ C = 0.1 * (w_1 * b_1 + w_2 * b_2 + w_3 * b_3)$$

Equations 2 to 6, provide the results which are optimally fit for the ANN model. The proposed model includes one input layer, one hidden layer, and one output layer respectively. The number of nodes in each layer includes 2, 3, and 1 respectively. The learning rate determines the accuracy of artificial neural networks with bionic technology. When the learning rate increases, the artificial neural network learns faster. The performance study sets the learning rate variations as shown in table 2 for the optimal global solution. ANN model is used where the population size varies from 100 to 500 with an increment of 50.

### 3.4 Neural model training

In this phase, The prepared data sets divide in to two parts 80% data set used for the training of the model shown in the block diagram and 20% data set use for the testing of the training model using variations of iterations. The training process include following steps 1. Initialization of the bias parameter and connection weights. 2. Training the model using data sets generated using whale optimization base scheduling in cloud computing 3. Validation must be performed using 20% data sets for the accuracy test and efficiency improvement of the proposed model. 4. Once the model is validated then the network runs to predict the values of the expected outputs. The linear and non-linear activation function use for the activation of the neurons in a scalable cloud computing environment. The complexity of the neural network classifier i.e. number hidden layers and types of activation functions depends on complexity of the datacenter node. Hence training step plays a prominent role in a scalable cloud computing environment.

### 3.5 Error backpropagation and correction

In this phase our objective is mitigate the difference between desired output and target output. Hence in this supervised learning process the error difference must be minimum. Smaller the difference between target and desired output better the accuracy of the

**Table 1.** Simulation Parameters of BB-BC-ANN & Hyper-parameters of the Neural Network

Hyper-parameters	Values
Number of iterations	100,150, 200, 250, 300, 350, 400, 450, 500
Layers	Input layer:1, Hidden layer: 1 Output layer:1
Learning Rate	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9
Mutation Rate	0.15
Crossover Probability	0.2
Population Size	100-500
Evolution Count	100
Count of Neurons (Input Layer)	2
Count of Neurons At Output Layer	1
Count of Hidden Layers	1
Count of Hidden Units	3
Activation Function (Hidden Layers)	Leaky ReLU Activation function=(fx) = $\{\alpha *x, \text{ if } (x \leq 0); x, \text{ if } (x > 0)\}$ $\forall \text{forall } \alpha = 0.01$

model in a complex, scalable distributed datacenter network environment. The error correction learning machine uses equations 8 and 9.

$$e_k(n) = d_k(n) + y_k(n) \quad (8)$$

$$\delta_k = \frac{1}{2} * e_k^2(n) \quad (9)$$

### 3.6 Task scheduling

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

### 3.7 Steps of Big-Bang Big-Crunch

The proposed model considers that a cloud is a heterogeneous combination of various cloud services that provide cloud resources with varying cost, resources, and performance. Cloud controller and task scheduling policy are responsible for finding a suitable resource on a service provider to complete the task in the least time, execution cost, and with a defined quality of service. The proposed model is meant to cover all the performance parameters mentioned above.

The proposed model is divided into 4 modules:  
Module 1: Initialization / Population generation  
Module 2: Fitness Evaluation  
Module 3: Crossover/Center of mass + Mutation  
Module 4: Big Crunch

#### Phase 1: Initialization / Population generation

In this initialization of population and other basic parameters are done, where the population is considered as a completed schedule consisting of tasks that are generated using Poisson distribution because the task's occurrence is completely random in the real world. The function of probabilistic distribution is given below in equation 10.

$$P(X = x) = \frac{\gamma^x e^{-\gamma}}{x!} \quad (10)$$

where  $x = 0, 1, 2, 3, \dots$

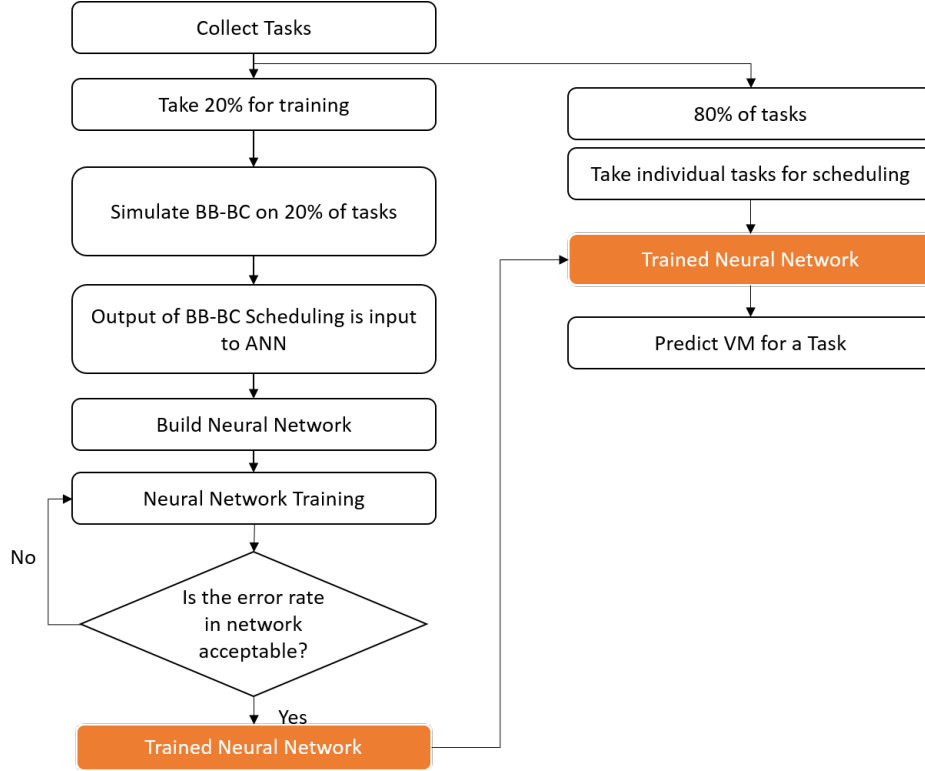
Where  $x$  is a natural number showcasing the population size i.e. the number of tasks.  $\lambda$  is the rate of requests arriving which is usually greater than zero. In this phase, 'p' pollutions are initialized with random VM's been allocated to the task. The populations finally constitute a predefined set of randomly generated tasks and each in a population is initialized with random VM id's. After this phase, the fitness value for each population is initialized which defined the performance and quality of service of each schedule.

#### Phase 2: Fitness evaluation

This phase starts with the evaluation of the fitness value of each population as given below. Fitness plays an important role to identify whether the schedule will take part in the next generation of evolution or not. Fitness in BB-BC is also referred to as the mass of the schedule. In the proposed model, the fitness function is the fitness of  $i^{th}$  population. The fitness function is the sum of cost and execution of each task over a VM where  $j$  states the number of tasks as shown in equation 11. Where  $\alpha + \beta = 1$

$$fitnessValue_i = \sum_{j=1}^{j=n} \alpha * NetworkCost_j + \beta * TaskCompletionTime_j \quad (11)$$

The task completion time variable is measured using equation 12 which Task\_Length is the number of instructions of a task, VM\_MIPS is the MIPS (millions of instructions per second) of VM



**Fig. 3.** Block Diagram of the Neural Network Training.

and PE is the number of processing elements.

$$task\_CompletionTime_i = \frac{Task\_Length_i}{VM\_MIPS_i * PE_i} \quad (12)$$

$$Networ\_Cost_i = Network\_delay_i \quad (13)$$

Equation 13 defines the cost of resources used during the execution of a task at a specific VM. In this phase, the mass of all the schedules is evaluated and then evolution takes place. Evolution is a set of steps repeated in every evolution which include selection, crossover, mutation, and the big crunch phase.

#### Phase 3: Crossover/Center of mass & Mutation

In this phase, the two best solutions are selected. The first solution is the one with the least cost and the second solution is the one with fitness value near the center of mass of all solutions. Multi-point crossover is selected to transfer more information from one chromosome to another chromosome where single-point crossover has less probability of information transfer. The center of mass or contraction operator finds out using equation 14. The point which represents the center of mass is denoted by  $x^c$ . Its values are calculated using the

formula in equation 14.

$$x^c = \frac{\sum_{i=1}^{i=n} \frac{1}{f_i} x_i}{\sum_{i=1}^{i=n} \frac{1}{f_i}} \quad (14)$$

Where  $x^c$  is a point within an n-dimensional search space generated in the first step. N is the population size,  $f_i$  is the fitness function value. The center of mass is represented by the point which is used to find the best solution. The candidate solution is assigned at the center of mass  $x^c$ . Then a solution around the center of mass is found by addition or subtraction of standard random number. The original point near the center of mass  $x^c$  is measured using equation 14 which finds the new candidate solution near the center of mass. In the experimental work, the fitness function is measured as shown in equation 11 and a new candidate solution is measured around the center of mass using equation 14. The shown in equation 15 represents the new candidate solution around the center of mass and  $x^c$  indicates the center of mass. Variable “l” is the upper limit of the parameter, r is a usual random number, and k is the iteration step

$$x^{new} = x^c + \frac{lr}{k} \quad (15)$$

The second-best solution is the solution near the



center of mass. Crossover: This phase takes the two best solutions from the selection phase and a multi-point crossover is performed in which swapping of random points between the two selected solutions is done. This phase aims to design a better solution from the two best solutions. Mutation: In this phase, the third-best solution discovered spreads its diversity in other populations using swapping based mutation of candidate solution in a search space with the other solutions.

**Phase 4: Big Crunch**

This phase plays an important role by removing the population with the worst fitness value i.e. the highest fitness value. Repeat the phases 2, 3, and 4 until one of the best solutions is left.

**4. Performance evaluation**

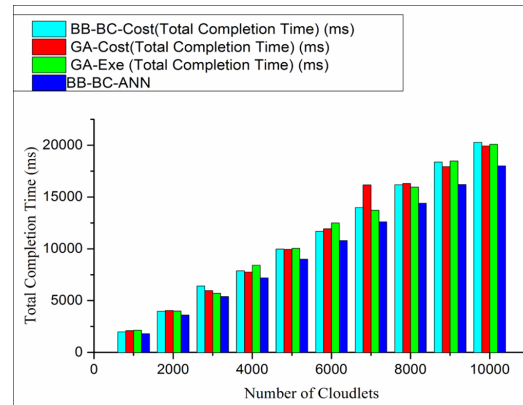
In this section the simulation setup and result using various simulation configurations are discussed. Simulation is performed using Cloudsim 3.0, The simulation uses workload traces for real time task simulation which is a SWF format workload file from parallel workload. Performance is measured using the scheduled time of the tasks, execution time, and operational cost paid for the resources at the infrastructure level. Table 3 shows the environmental setup of the cloud environment inside the datacenter.

**Table 2.** Datacenter Configuration Parameters

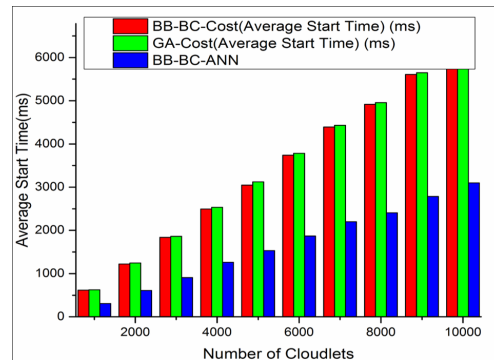
Datacenter ID	Secondary Storage (GB)	RAM(GB)	PE	Hosts	CORE
D1	100000	64	6	2	4
D2	100000	64	6	2	4
D3	100000	64	6	2	4
D4	100000	64	6	2	4
D5	100000	64	6	2	4

**4.1 Results using fabricated data sets**

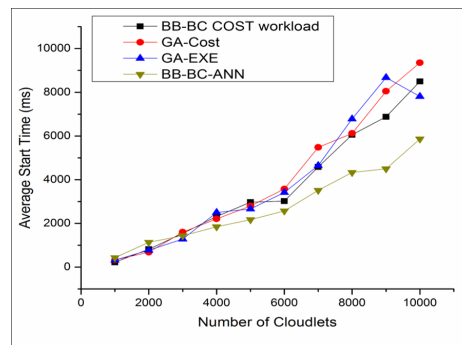
Figure 3 exhibits the comparisons among four different approaches. The performance is measured using ten different scenarios. The genetic approach with an artificial neural network provides an optimal solution. This model is the most effective for the ten different scenarios. Figure 4 exhibits the variations of the average start time on increasing the number of tasks. The neural network-based model provides optimal results.



**Fig. 4.** Total completion time Ccomparisons.



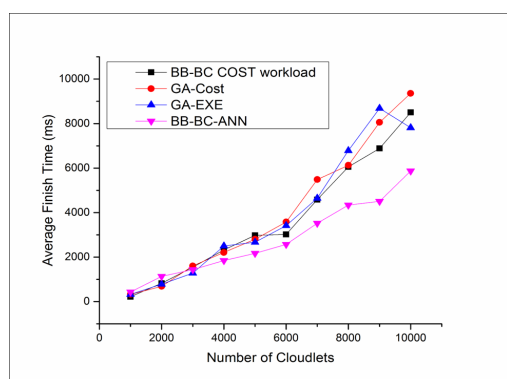
**Fig. 5.** Average start time comparisons.



**Fig. 6.** Average start time (ms) comparison with the cloudlets.

**4.2 Results using real workload file (The San –Diego Super Computer Center (SDSC) & Blue Horizon Logs)**

This work uses records to measure the performance of an IBM SP with 144 nodes and 8 processors per node. The log runs from April 2000 to January 2003. The number of jobs in the log file; H. 250, 440. Figure 5 shows the performance using

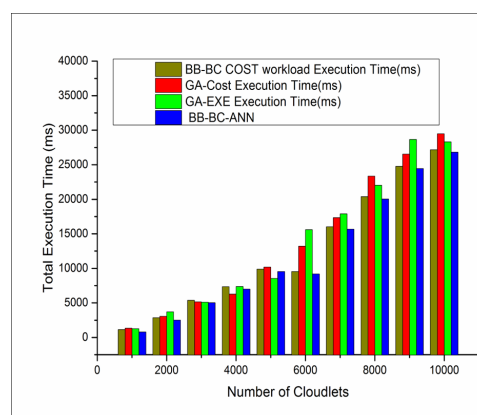


**Fig. 7.** Average finish time (ms) comparison with cloudlets.

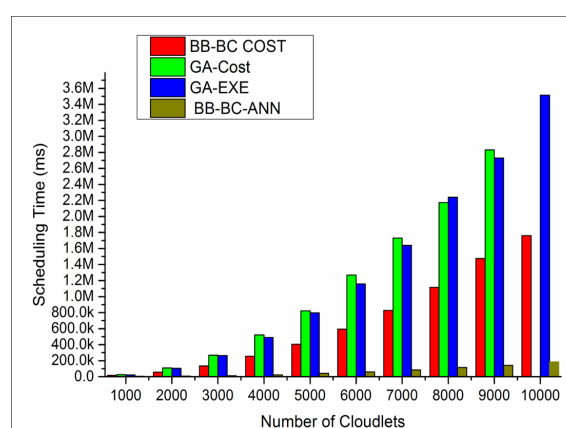
average start time variations with an increasing number of task requests. The proposed BB-BC-ANN model outperforms the existing bio-inspired approaches. Figure 6 exhibits the variations of the average finish time for the number of submitted tasks. The tasks are submitted on virtual machines using proposed BB-BC-ANN, BB-BC cost, and two other variants of the genetic approach. Figure 7 exhibits the variations of total execution time with the number of tasks. The tasks are generated using the Real Workload File (The San Diego Supercomputer Center (SDSC) Blue Horizon logs). The performance metric total execution time is improved by increasing the number of user requests. The proposed BB-BC-ANN model is promising and good optimal for the ten different scenarios. Figure 8 shows the scheduling timing in a scalable cloud environment. Cloudsim 3 implements four meta-heuristic methods inspired by nature. The proposed BB-BC-ANN model shows better results using scheduling time as a performance indicator. The scheduling time is improved with the number of tasks or user requests generated using Real Workload File.

## 5. Conclusion

In this work, an efficient task scheduling scheme was presented for task scheduling in cloud infrastructure. The proposed BB-BC-ANN model outperforms the genetic approach and Big-Bang Big-Crunch cost-aware techniques. The training, validations, and predictions are performed using real-time data sets from standard workload file and fabricated data sets. The results exhibit that the proposed BB-BC-ANN model outperforms the existing nature-inspired techniques. The results show that the proposed BB-BC-ANN model



**Fig. 8.** Variations of total execution time versus the number of cloudlets.



**Fig. 9.** Number of cloudlets (number of requests using workload logs) versus scheduling time (ms).

improves the performance of cloud applications in the real-world with the execution of the task in the least scheduling time and execution time (ms). In future work, the proposed model will be used for virtual machine scheduling on the host and virtual machine migration approaches for better utilization of resources to improve the running cost of cloud applications with optimal resource utilization.

## References

- Amiri, M. A., Conoscenti, C. and Mesgari, M. S. (2018). Improving the accuracy of rainfall prediction using a regionalization approach and neural networks. *Kuwait Journal of Science*, **45**(4); pp. 66–75.
- Ang, T. F., Por, L. Y., & Liew, C. S. (2017). Dynamic pricing scheme for resource allocation in multi-cloud environment. *Malaysian Journal of Computer Science*, **30**(1); pp. 1-17.
- Arabnejad, V., Bubendorfer, K., & Ng,

- B. (2019).** Budget and Deadline Aware e-Science Workflow Scheduling in Clouds. *IEEE Transactions on Parallel and Distributed Systems*, **30**(1); pp. 29–44.
- Arunarani, A. R., Manjula, D., & Sugumaran, V. (2019).** Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, **91**; pp. 407–415.
- Beegom, A. A., & Rajasree, M. S. (2014, October).** A particle swarm optimization based pareto optimal task scheduling in cloud computing. In *International Conference in Swarm Intelligence*; pp. 79-86.
- Beheshti, Z., & Shamsuddin, S. M. H. (2013).** A review of population-based meta-heuristic algorithms. *Int. J. Adv. Soft Comput. Appl.*, **5**(1); pp. 1-35.
- Beloglazov, A., Abawajy, J., & Buyya, R. (2012).** Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future generation computer systems*, **28**(5); pp. 755-768.
- Buyya, R. (2013).** Introduction to the *IEEE Transactions on Cloud Computing*. *IEEE Transactions on Cloud Computing*, **1**(1); pp. 3–21.
- Dabbagh, M., Hamdaoui, B., Guizani, M., & Rayes, A. (2015).** Energy-Efficient Resource Allocation and Provisioning Framework for Cloud Data Centers. *IEEE Transactions on Network and Service Management*, **12**(3); pp. 377–391.
- Domanal, S. G., & Reddy, G. R. M. (2014, January).** Optimal load balancing in cloud computing by efficient utilization of virtual machines. In *2014 Sixth International Conference on Communication Systems and Networks (COMSNETS)*; pp. 1-4. *IEEE*.
- Domanal, S. G., & Reddy, G. R. M. (2018).** An efficient cost optimized scheduling for spot instances in heterogeneous cloud environment. *Future Generation Computer Systems*, **84**; pp. 11-21.
- Faragardi, H. R., Saleh Sedghpour, M. R., Fazliahmadi, S., Fahringer, T., & Rasouli, N. (2020).** GRP-HEFT: A Budget-Constrained Resource Provisioning Scheme for Workflow Scheduling in IaaS Clouds. *IEEE Transactions on Parallel and Distributed Systems*, **31**(6); pp. 1239–1254.
- Genez, T. A. L., Bittencourt, L. F., Da Fonseca, N. L. S., & Madeira, E. R. M. (2019).** Estimation of the Available Bandwidth in Inter-Cloud Links for Task Scheduling in Hybrid Clouds. *IEEE Transactions on Cloud Computing*, **7**(1); pp. 62-74.
- Gu, J., Hu, J., Zhao, T., & Sun, G. (2012).** A new resource scheduling strategy based on genetic algorithm in cloud computing environment. *Journal of computers*, **7**(1); pp. 42-52.
- Jin, X., Zhang, F., Wang, L., Hu, S., Zhou, B., & Liu, Z. (2015).** Joint optimization of operational cost and performance interference in cloud data centers. *IEEE Transactions on Cloud Computing*, **5**(4); pp. 697-711.
- Juarez, F., Ejarque, J., & Badia, R. M. (2018).** Dynamic Energy-Aware Scheduling for Parallel Task-Based Application in Cloud Computing. *Future Generation Computer Systems*, **78**(1); pp. 257-271
- Kalra, M., & Singh, S. (2015).** A review of metaheuristic scheduling techniques in cloud computing. *Egyptian informatics journal*, **16**(3); pp. 275-295.
- Kaur, K., Chhabra, A., & Singh, G. (2010).** Heuristics based genetic algorithm for scheduling static tasks in homogeneous parallel system. *International Journal of Computer Science and Security (IJCSS)*, **4**(2); pp. 183-198.
- Kruekaew, B., & Kimpan, W. (2014, March).** Virtual machine scheduling management on cloud computing using artificial bee colony. In *Proceedings of the International Multi Conference of engineers and computer scientists*. **1**; pp. 12-14.
- Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006).** Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE transactions on evolutionary computation*, **10**(3); pp. 281-295.
- Liu, A., & Wang, Z. (2008, October).** Grid task scheduling based on adaptive ant colony algorithm. In *2008 International conference on management of e-commerce and e-government*; pp. 415-418.

- Liu, X. F., Zhan, Z. H., Deng, J. D., Li, Y., Gu, T., & Zhang, J. (2018).** An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing. *IEEE Transactions on Evolutionary Computation*, **22**(1); pp. 113–128.
- Lu, X., & Gu, Z. (2011, September).** A load-adaptive cloud resource scheduling model based on ant colony algorithm. In *2011 IEEE international conference on cloud computing and intelligence systems*; pp. 296-300. IEEE.
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013, June).** Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the 30th International Conference on Machine Learning*, 30(1); pp. 3-11.
- Mishra, S. K., Puthal, D., Sahoo, B., Jena, S. K., & Obaidat, M. S. (2018).** An adaptive task allocation technique for green cloud computing. *The Journal of Supercomputing*, **74**(1); pp. 370-385.
- Mosa, A., & Paton, N. W. (2016).** Optimizing virtual machine placement for energy and SLA in clouds using utility functions. *Journal of Cloud Computing*, **5**(1); pp. 1-17.
- Parallel Workloads Archive:  
<https://www.cse.huji.ac.il/labs/parallel/workload/>
- Rawat, P. S., Dimri, P., Kanrar, S., & Saroha, G. P. (2020).** Optimize Task Allocation in Cloud Environment Based on Big-Bang Big-Crunch. *Wireless Personal Communications*, **115**(3); pp. 1711–1754.
- Sahni, J., & Vidyarthi, P. (2018).** A Cost-Effective Deadline-Constrained Dynamic Scheduling Algorithm for Scientific Workflows in a Cloud Environment. *IEEE Transactions on Cloud Computing*, **6**(1); pp. 2–18.
- Senthil Kumar, A. M., & Venkatesan, M. (2019).** Multi-Objective Task Scheduling Using Hybrid Genetic-Ant Colony Optimization Algorithm in Cloud Environment. *Wireless Personal Communications*, **107**(4); pp. 1835–1848.
- Shojafar, M., Javanmardi, S., Abolfazli, S., & Cordeschi, N. (2015).** FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method. *Cluster Computing*, **18**(2); pp. 829–844.
- Singh, P., Dimri, P., Saroha, G. P., & Barthwal, V. (2016).** A Load Balancing Analysis of Cloud Base Application with different Service Broker Policies. *International Journal of Computer Applications*, **135**(10); pp. 11-15.
- Singh, Poonam, Dutta, M., & Aggarwal, N. (2017).** A review of task scheduling based on meta-heuristics approach in cloud computing. *Knowledge and Information Systems*, **52**; pp. 1–51.
- Singh, S., & Chana, I. (2016).** A survey on resource scheduling in cloud computing: Issues and challenges. *Journal of grid computing*, **14**(2); pp. 217-264.
- Soroush, H. M. (2013).** Scheduling stochastic jobs on a single machine to minimize weighted number of tardy jobs. *Kuwait Journal of Science*, **40**(1); pp. 123-147, 2013
- Sreelatha, K. S. M. (2017).** W-Scheduler: whale optimization for task scheduling in cloud computing. *Cluster Computing*, **22**; pp. 1087–1098.
- Sri, L., & Narayanan, B. (2017).** Speculation resource provisioning in high-performance computing. *Kuwait Journal of Science*, **44**(1); pp. 58-63, 2017
- Wei, L., Foh, C. H., He, B., & Cai, J. (2018).** Towards Efficient Resource Allocation for Heterogeneous Workloads in IaaS Clouds. *IEEE transactions on Cloud Computing*, **6**(1); pp. 264–275.
- Wu, Q., Zhou, M., Zhu, Q., Xia, Y., & Wen, J. (2020).** MOELS: Multiobjective Evolutionary List Scheduling for Cloud Workflows. *IEEE Transactions on Automation Science and Engineering*, **17**(1); pp. 166–176.
- Xiao, Z., Song, W., & Chen, Q. (2013).** Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, **24**(6); pp. 1107–1117.
- Yu, J., & Buyya, R. (2006).** Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming*, **14**(3); pp. 217-230.

**Zhan, Z. H., Liu, X. F., Gong, Y. J., Zhang, J., Chung, H. S. H., & Li, Y. (2015).** Cloud computing resource scheduling and a survey of its evolutionary approaches. *ACM Computing Surveys (CSUR)*, 47(4); pp. 1-33.

**Zhao, C., Zhang, S., Liu, Q., Xie, J., & Hu, J. (2009, September).**Independent tasks scheduling based on genetic algorithm in cloud computing. In *2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*; pp. 1-4. IEEE.

**Zhou, Z., Li, F., Zhu, H., Xie, H., Abawajy, J. H., & Chowdhury, M. U. (2020).** An improved genetic algorithm using greedy strategy toward task scheduling optimization in cloud environments. *Neural Computing and Applications*, 32(6); pp. 1531–1541.

**Submitted:** 02/05/2020

**Revised:** 11/01/2021

**Accepted:** 12/01/2021

**DOI:** 10.48129/kjs.v48i4.9664