DRL Based Multi-objective Resource Optimization Technique in a Multi-cloud Environment

Ramanpreet Kaur[1,2](✉), Divya Anand[3], and Upinder Kaur[4]

[1] Department of Computer Application, Lovely Professional University, Phagwara, Punjab, India

[2] Department of Computer Science, Baba Farid College, Bathinda, Punjab, India
ramaninsa1990@gmail.com

[3] Department of Computer Science and Engineering, Lovely Professional University, Phagwara, Punjab, India
Divyaanand.y@gmail.com

[4] Department of Computer Science and Engineering, Akal University, Talwandi Sabo, Punjab, India
upinder_cs@auts.ac.in

Abstract. The concept of multi-cloud becomes interesting progressively to cloud users because of its high response time, flexibility, high throughput, and reliability. But at the ground level, the concept of multi-cloud creates many challenges for researchers. The request of users and the multi-cloud environment is heterogeneous now a day. To work in this kind of environment required an intelligent system. Researchers are doing well in this field to make the whole process very flexible by providing an intelligent environment. The proposed multi-objective resource optimization deep reinforcement learning (MOROT-DRL) model uses the Q-learning technique of Deep Reinforcement Learning (DRL) to allocate resources in a multi-cloud environment. It includes a service analyzer for analyzing the requests and MET(Minimum Execution Time)algorithm used for scheduling the task according to execution time and then enhanced flower pollination allocate the optimized resources for the demanded request. The comparison of the proposed model is done with simulation results of MOROT and neural network model and also implemented on GoCS real dataset of google. The proposed model gives better results when compared based on energy, $CO_{2,}$ and cost.

Keywords: Multi-cloud · Deep reinforcement learning · Resources allocation · Cyber shake seismogram workflow · Task scheduling Enhanced Flower Pollination

1 Introduction

Based on daily consumer need the demand for various services like SaaS, PaaS, and IaaS, promoting everyone to in the environment having multiple cloud. The concept of multi-cloud is used when the services are fulfilled from the various cloud or the services are moved from one cloud to another cloud. The author presents a taxonomy related to

P. Pareek et al. (Eds.): IC4S 2023, LNICST 536, pp. 106–121, 2024.
https://doi.org/10.1007/978-3-031-48888-7_9

multi-cloud mentioned that the main actor that works in multi-cloud is Cloud Service Provider (CSP) [1]. Cloud computing provides high performance and a large number of services to the cloud user based on pay–per–usage. The cloud resource is distributed in various locations and connected according to a geographical area. These services are not allocated physically to the user but rather can be used on a rent based. Cloud user works with multi-cloud without knowing the physical location of the cloud which is fulfilling their request, same as virtual machines are also unknown to the user's location. Service Level Agreement (SLA) is an important part between cloud users and cloud providers because it deals with important parameters such as quality, pricing, security, etc. [2]. So to fulfill the quality of services based on SLA, there is a requirement for efficient techniques of resources management. Various researchers are doing work in this field by providing various features of optimized resource allocation techniques.

Task scheduling is also very important with optimized resource allocation. Task scheduling means scheduling the incoming task in such a manner that efficient resources can be allocated. Various task scheduling approaches are doing well in this field. The suitable technique can be adopted based on the suitable requirement of cloud users or the resources allocation technique. Lots of work is done for resource allocation and task scheduling in cloud computing. But the same concept becomes very complicated in the case of a multi-cloud environment.

As the services and users in multi-cloud are increased, the concept of task scheduling and resource allocation becomes a challenge for researchers. DRL technique of machine learning performs the best role in every IT-related field. It can make a very complex decision which was not possible in previous machine learning. In the cloud and multi-cloud, DRL also performs an important role in traffic, identification, future prediction, task scheduling, and resource allocation (Fig. 1).

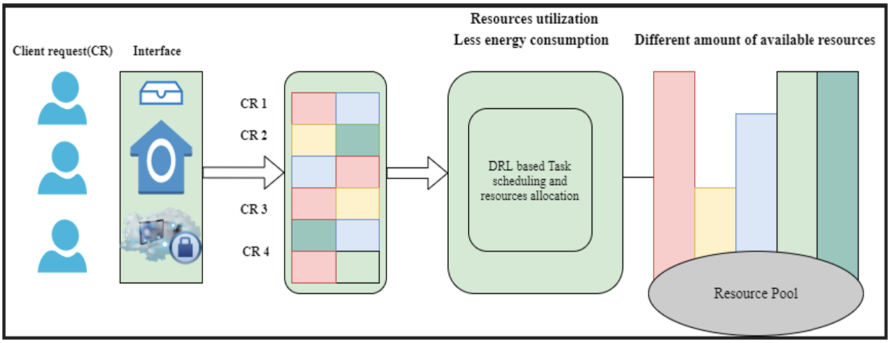

Fig. 1. Representation of resources allocation in multi cloud environment.

The performance of cloud or multi-cloud can be measured based on performance indicators or some parameters such as makespan, reliability, throughput, time, cost, power, and carbon emission. Deep reinforcement learn in intelligence elegant techniques are performing an amazing role to improve resource allocation in a multi-cloud environment without any SLA violation.

Poor resource allocation or under-utilization of resources is responsible for high consumption of energy, cost, and some other precious parameters. High energy consumption leads to a high ratio of CO_2 and both deeply affects social life. The balance between both is a very challenging concept. The future of any country depends on a healthy citizens. The deep reinforcement learning technique intelligently allocates resource to the user from more than one cloud to fulfil their requests having all benefits as energy efficiency, cost and CO_2 reduction. The enhanced flower pollination algorithm (EFPA) works in the particular cloud for resource allocation based on local and global optimization techniques for energy efficiency. MET algorithm and cybershake seismogram workflow segregate the incoming requests and prepare a task queue for EFPA so that requests can be processed based on minimum execution time. This model is evaluated with cloudsim simulator. The remainder part of this paper includes as following. Section 2 is the related work and parameter-based study of task and resource allocation in the multi-cloud. Section 3 is the explanation of the proposed model having a flowchart and algorithms. The experiment results discussion and comparison of proposed work is in Sect. 4.

2 Related Work

This section gives a basic review of various task scheduling and resource allocation techniques. Authors doing well to reduce various issues in the field of cloud and multi-cloud environments.

Predicting the future user's requirements in the cloud helps to reduce violations in service level agreements. But it is very difficult to predict the future requirement in the case of multi-cloud. Future prediction helps the cloud provider to allocate quality of service to the user. The author proposed a hybrid approach for future requirement prediction. This approach uses lazy learning, modified K-medoids, and lower bound dynamic time warping. This proposed approach gives better prediction when compared with others [3]. Resources management in multi-cloud needs a unique interface and wrapper for every service. The author proposed an approach that is adopted by deployable services in terms of open sources available platforms. This interface is different at run time, design, and deployment stages. The focus of this paper is to give an open-source, module-based solution that can be easily used [4]. The author presented how Cloud MF makes techniques of model-driven and ideas for minimizing the vendor lock-in and helps for allocations applications of multi-cloud. The Cloud ML (Cloud Modelling Language) permits to provision and deployment of applications in models of cloud provider-independent [5]. The author developed an optimized approach to minimize micro services repair, latency overhead of allocating containers on the cloud and reduce services cost. As micro-service are arranged in a container and that container will be allocated to VM but how to allocate the container on a suitable VM and allocate VM on a suitable cloud is a challenging issue. The author implements the NSGA-II genetic algorithm and compares it with the Greedy First-Fit algorithm; the implemented algorithm gives 300% improvement as compared to others [6].

Services provided by the cloud make every task very flexible as a business also moved toward the cloud and getting more benefits. To work on a single cloud is very efficient. But difficult in the case of multi or cross-cloud. To handle different instances of business processes near customers can be beneficial; the author presents a novel architecture of the environment of multi-cloud business provision. This architecture involves components to handle the monitoring and adaption of the business processes in a multi-cloud environment. This framework explains all about services such as Iaas, PaaS, monitoring, adoptions, etc. that can help to do business processes in a better way [7]. The author presented Replica aware task scheduling method to reduce the response delay of services. According to this algorithm, transferring computation and transfer data are combined. Resources matching are accomplished according to the availability of nodes. Failed or non-local data is replicated in advance to the targeted node. According to the cache placement algorithm, the next execute task is predicted. The experiment result shows that our proposed method performs better as compared to the benchmark algorithm in terms of node prediction and response time [8].

An integer linear programming model is developed to handle the scientific workflow in a multi-cloud environment. This helps to reduce financial costs by encountering the deadline requirement of the user. In this proposed model the resource limit imposed by the cloud provider and cost is calculated on an hourly basis. The experiment results show that change in deadline and workflow affect cost i.e., greater in CPU intensive workflows rather than other elasticity values remain always a constraint in work under long deadlines. A short deadline has a high cost. The comparison of the proposed model is done with the MIP-CG, MCPCPP, and IP-FC. The result shows that the proposed model is suitable for all deadlines; in the future, the makespan and total cost should be considered [9].

The author proposed Multiple-replica integrity auditing schemes for secure data storage on the cloud. A Cloud users are continuously taking data storage services on the cloud free from cost burden. The Author also mentioned open issues and research directions [10]. The author presents a hybrid formal verification approach for accessing high-quality service composition in the environment of multi-cloud by reducing the no. of the cloud provider. The proposed approach is helpful for checking the user request, services selection, and multi-cloud composition. Results show that this method reduces memory consumption [11]. This paper investigates resource management in a multi-cloud environment. The author also investigates the user's demand for applications in a multi-cloud environment. Definition and resources classification in the multi-cloud environment and three taxonomy of multi-cloud are mentioned. Future trends and challenges also point out [12]. In this paper, the author focus on scheduling techniques that handle challenges in inter-cloud and also presents basic concerns and task scheduling related to multi-cloud. All scheduling techniques are categories based on some parameters. After containing the survey, the author mentions that security and load balancing is an important concern that should be considered in the future [13]. This author proposed a cloud-enabled workflow science gateway. This paper includes all the principles of integrating the cloud system with a science gateway. It integrates the WS-PGRADE/g USE and cloud broker platform. This integration is used in the CloudSME project where 20 companies port the simulation application on the cloud. The proposed method provides cloud access flexibility and the user can access all clouds integrated with this gateway [14]. During the use of multi-cloud, the user has to face some challenges such as provisioning, elasticity, portability, and availability. To handle these challenges the author presented the so cloud framework that is deployed on 10 cloud providers' complete architecture and interaction between all components of the so Cloud framework discussed in this paper. With this approach, the user can get high availability [15].

The author focuses on the problem of VM placement for reducing cost and saving energy in a heterogeneous environment of multi-cloud. The author proposed a mimetic algorithm for VM placement based on cost-efficient to solve this problem i.e. called grouping genetic. The proposed algorithm reduces running PM and consumption of energy by the geographical distribution of the data center. Hill climbing is also used do searching in local to maximize the speed and run time of the genetic algorithm. Comparison of the proposed model is made with three other recent researches and found that the proposed model performs better to reduce cost and energy [16]. In this paper, the author developed normalized hybrid service brokering With Throttled Round Robin Load Balancing (NHSB_TRB) to provide cost-effective services to the user. This approach produces a normalized value of optimized cost. The data center based on cost is selected for distributing the load. The weighted threshold is used to distribute the load on the data center and the round-robin load balancing approach distributes the load on VM. The experiment result shows that the proposed model improves response time, monetary cost, and processing time of data center up to 17.39%, 7.06%, and 31.35 when compare with ORT_RR, CDC_RR, ORT_THR, and ORT_ES approaches [17]. Table 1 compares task scheduling and resources allocation techniques in a multi-cloud environment based on a parameter such as Makespan, cloud/resource, utilization, scalability, cost, Response Time, Energy efficiency, and Co2 reduction.

The above table shows that not a single technique works collectively on cost, energy, and $Co_{2.}$ So the same should be considered in research work.

3 Proposed Methodology

With the time-varying demands of construction, optimization, and user requests, a Cloud computing system cannot be considered self-sufficient. The internal environment for agent state and decision-making are also temporal shifting like Cloud computing's income ratio, which varies in different periods during a single day [31]. A DRL framework with time-varying external stimuli is used to view the decision-making process as an ensemble. Each mapper's information is as follows: Extrinsic or internal stimuli are inputs to the mapper of time-varying, whereas the output is a change in the agent's state or state-changing. It is a mapper of stimulus evolution in agent and state, where the stimulus force is input and the set of agent-state at real-time is output, as the agent and state are usually changing with stimuli.

When it comes to managing resources in a cloud computing environment, resource allocation (RA) is one of the methods available. When it comes to establishing the optimal balance between VM and PM in the cloud data center, Infrastructure as Service providers confronts a significant difficulty. This is known as finding the optimal allocation for VMs and PMs in terms of the number of resources they require [32]. There are two components to resolving the issue of VM allocation: first, the acceptance of new requests for VM provisioning and the placement of the VMs on a PM, and second, the optimization of existing VMs (Fig. 2).

Table 1. Task scheduling and resource allocation techniques in a multi-cloud environment.

Ref. No.	Technique	Makespan	resources utilization	Scalability	Cost	Response Time	Energy efficiency	Co_2 Reduction
SK Panda [18]	AMinB, AMaxB, and AMinMaxB task scheduling algorithm	✔	✔	×	×	×	×	×
Lijin P [19]	Game theory-based resources allocation algorithm	✔	×	×	×	×	×	×
A. Pietrabissa [20]	Q-learning(Policy reduction and state aggregation strategy)	×	×	✔	×	×	×	×
SK Mishra [21]	Energy-Aware Task Allocation in Multi-Cloud Network	×	×	×	×	×	✔	×
J.Carvalho [22]	Simple Adaptive weighting method and multi-choice knapsack problem	×	✔	×	✔	✔	×	×
P.Antonio1 [23]	SARSA(λ) and Q-learning	×	×	×	×	×	×	×
S.Kang [24]	DSS	×	×	×	×	×	×	×
Z.Chen [25]	OWS-A2C	×	✔	×	✔	×	×	×
SK Panda [26]	MCC,MEMAX,CMMN Algorithm	✔	✔	×	×	×	×	×
M.Farid [27]	FR-MOS	✔	✔	×	✔	×	×	×
T.Subramanian [28]	Novel cloud brokering architecture	×	×	✔	×	×	×	×
C. Thirumalaiselvan [29]	ELB, high priority scheduling algorithm, and rate-based scheduling algorithm	✔	×	×	×	✔	✔	×
N.Grozev [30]	Rule-based domain-specific model	×	×	×	×	×	×	×

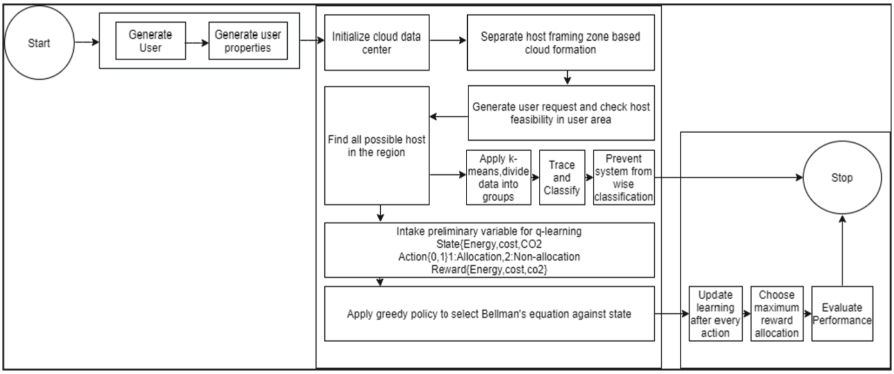

Fig. 2. Workflow of DRL in a cloud environment.

The decision mapper is in charge of computing the next action based on the agent's present state, and the action taken at the next opportunity is the output. To put it in layman's terms: The environment is fed by actions provided by a mapper and grows as a result of those actions. Environment's output is fed into mapper of feedback, while agent's internal stimuli are fed into mapper of time-varying [33–35]. Replay storage is used to store long-term input in preparation for future use, while timely feedback is taken from the environment. The settings of the decision-maker will be updated as a result of both long-term and real-time feedback. Generalized RL built on the integration of mappers. Programs are often used to model time-varying, stimulus evolution, and environmental conditions in some research. Neural networks can be used to build a decision and feedback mapper. The feedback mapper can be implemented as a neural network to calculate the loss function of the neural network in the decision mapper since it aims to update the parameters of the decision-maker. In computing, a VM (virtual machine) is an emulation of a certain computing system that is used to simulate another system. Virtual machines are capable of running since they are based on the computer architecture and functionalities of a real or physical computer. These systems can be implemented using specialized hardware and software, or they can be implemented using a combination of both [36]. Virtual machines can be divided into several categories based on how closely they resemble their real-world counterparts in terms of capability (Fig. 3).

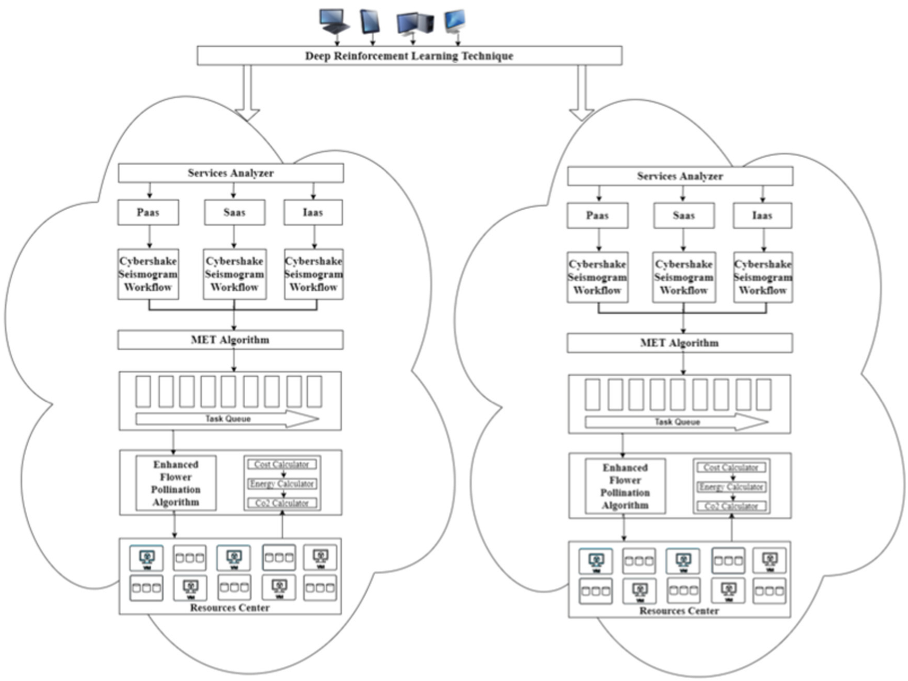

Fig. 3. Multi-objective resource optimization deep reinforcement learning (MOROT-DRL) model

As a result, system virtual machines can serve as a complete substitute for the targeted virtual machines, as well as providing the amount of functionality required to operate an operating system (also known as full virtualization VMs). While this is true, a process virtual machine provides an abstracted and platform-independent execution environment for a single computer application running on a variety of platforms. The centralized cloud resource manager is in charge of managing the resources in cloud computing. Cloud data center (DC) resources are made available to cloud consumers using virtual machines (VMs), which are based on physical machines (PMs). To maximize resource use while minimizing energy consumption, cloud computing Infrastructure as a Service (IaaS) providers must implement dynamic resource management techniques in their cloud DCs. Because of business considerations, the resource management strategies and algorithms used in public clouds are not revealed. The proposed energy-efficient resource allocation mechanism is comprised of a single central scheduling point (CSP) and N cloud users. CSP managed many heterogeneous resources like memory, processing units, network bandwidth, and so on in the form of virtual machines (VMs). When these virtual machines (VMs) were requested by cloud customers to complete their activities, the cloud provider allocated them based on the preferences of the cloud consumers. The primary purpose of this proposed Enhanced flower pollination algorithm is to reduce the amount of energy consumed during work scheduling in the cloud environment, as well as to reduce the number of task scheduling issues in cloud computing.

Algorithm 1: Enhanced flower pollination algorithms
1: Start
2: Input
Data center structure D_s
Size of Population S_m
Total number of takes $S_{m,\ task}$
Number of iterations i_{max}
Number of Virtual machines $S_{m,\ vm}$
3: Output
Optimal Solution Y_a
4: Calculate the global task queue
$Y_q^{j+1}=Y_q^j+\Upsilon(\Psi)(S-Y_q^j)$
5: Calculate the local task queue
$Y_q^{j+1}=Y_q^j+ɯ(Y_i^j-Y_m^j)$
6: finally find out the Optimal Solution
Y_a, a=1,2,3,………S_m
i = i+1;
7: Update and repeat the Optimal Solution
8: Select the best solution
9: Stop

Increased energy use results in an increased operating expense. The most pressing issue is the increase in excessive carbon emissions (CO2). It has a greater impact on the environment. This limited supply must be put to good use. The most critical step is to reduce the use of energy and power. It is important to avoid wasting resources. This is referred to as energy conservation. The efficient use of resources can be improved by utilizing virtualization technologies. The dynamic consolidation of virtual machines (VMs) is made possible by virtualization technology. Cloud service providers can host several virtual machines (VMs) on a single physical server, i.e. virtualization. One technique to reduce power usage is to turn off nodes that are not in use.

E_{CE} is a metric that measures how much energy is expended while a job is in the process of being set up for execution, such as copying the data needed for the task to run. The amount of CPU energy used to carry out the task in the designated VM is therefore considered as an E_{PE}, which is directly related to the amount of CPU energy utilized. Thus, the total amount of energy consumed by the user jobs in the Task Set may be calculated using Eq. 1.

$$E_{Total=\sum E_{VM}} \tag{1}$$

Here E_{VM} is known as

$$E_{VM} = E_{CE} + E_{PE} \tag{2}$$

Algorithm 2: Task Scheduling Algorithm for Deep reinforcement learning

1: Start
2: evaluate the resources information of task
3: Set E_{VM} task.
4: EvaluateE_{total}
$E_{Total=\sum E_{VM}}$
$E_{VM} = E_{CE} + E_{PE}$
5: Do till all task mapped
6: Earliest completion time and resources of all task are calculated
7: Set resource's ready time
8: According completion time set all resources
9: Do for all R
10: calculate highest Completion Time of Ti
If Maximum Completion Time <makespan
Compute makespan = max(CT(R))
11: Figure out task having minimum completion time.
12: Find out T_i with minimum ET
13: Reschedule task T_i according to produced resources.
14: Change ready time for those resources
15: End

4 Results and Discussion

To evaluate MOROT-DRL model, 'cloudsim' cloud computing simulation and Q-learning technique of DRL implemented for services analyzing. Go CS dataset is used as real cloud data set and comparison is also made on some well known websites dataset. The simulation program is written in Java and deployed on an HP i7 processor with 16GB RAM. The apache-commons mathematics library is used to generate the power versus throughput regression model for the servers. The Apache Net Beans used as a tool to open JDK 8, powered with an open V9 Java virtual machine used to run our code. These types of resources were considered for the testing CPU, memory & disk. A cloud data center information and customer configuration information are given in Tables 2 and 3.

Table 2. Data center information and customer configuration information.

Sr. No.	Characteristic	Value
1	Number of data centre	10
2	Number of hosts	200
3	Available bandwidth	100–7500 Hzs
4	Available Core	4
5	Capacity per core	3 octa engine
6	Engine Type	Multi
7	Engine Propagation	Quad Core
8	Process Utilization Minimum	1 Hzs
9	Single Core score	14323
10	Multi Core Score	14883
11	Engine Ram	2 GB

In this paper,a virtual environment is simulated to check the efficiency of the proposed method in terms of resources allocation.

The user requests the resources from the data center according to the requirement of the task. Here the CR is the Client request which is fulfilled with the help of a virtual machine VM. CR requests many resources at the same time.

$\mathbf{A}_{\mathrm{i}} \subset \mathbf{CR}$ and $x_s^1, y_s^1, z_s^1 \subset \mathbf{A}_{\mathrm{i}}$,
$x_s^1, y_s^1, z_s^1 \subset \mathbf{A}_{\mathrm{i}} \subset \mathbf{CR} \Rightarrow x_s^1, y_s^1, z_s^1 \subset \mathbf{CR}$,
When users request only one resource it can be written as Eqs. (3) and (4):

$$\mathbf{CR}^1 = \mathbf{A}_{\mathrm{i}}, \tag{3}$$

I mean 1 and $\mathbf{CR}^1$ means user request only 1 resource.

$$\mathrm{A} = (x_s^1 + y_s^1 + z_s^1) \tag{4}$$

Table 3. Notation in mathematical analysis.

Symbol	Definition
VM	Virtual Machine
CR	Client Request
A_i	Component of VM
X	Represents CPU
Y	Represents Memory
Z	Represents Storage
I	Number of Resources
S	Measuring Capacity
PM	Physical Machine
St_i	Starting Time of each VM_i
Et_i	Execution Time of each VM_i
RU^{DC}	Resource Utilization of data center
RU	Resource Utilization
TEC	Total Energy Consumption
EU	Energy Utilization
P	Power Consumption
DC_U^{Energy}	Energy Consumption of Data Center
T	Total Power Generated
S_k	Power Generated by Source k
ef_k	Emission Factor Related to k

When a user demands more than one resource it can be expressed as Eqs. (5) and (6).

$$\begin{aligned}\mathbf{CR}^{n} &= \sum\nolimits_{i=1}^{n} = \mathbf{A}_{i} = A_1 + A_2 + A_3 + \ldots A_n \\ &= (x_s^1 + y_s^1 + z_s^1) + (x_s^2 + y_s^2 + z_s^2) + \ldots + (x_s^n + y_s^n + z_s^n)\end{aligned} \tag{5}$$

$$\mathbf{CR}^{n} = \sum\nolimits_{i=1}^{n} (x_s^1) + \sum\nolimits_{i=1}^{n} (y_s^1) \sum\nolimits_{i=1}^{n} (z_s^2) \tag{6}$$

Energy and Resource Utilization Model

VM = $\{vm_i, i = 1, 2, \ldots, n\}$ are virtual machine allocated to Physical machine PM.

PM = $\{PM_j, j = 1, 2, \ldots, m\}$ are physical machines.

Three PM resources considered as physical memory (RAM), storage and processor (CPU).

So dimension d = 3.

The total time used during VM allocation is $S_{ti} + E_{ti}$.

where $A_{i,s}$ is resource capacity (x_s^1, y_s^1, z_s^1) requested by the $\mathbf{VM}_i(1, 2, \ldots, n)$ and $B_{i,s}$ resource capacity (x_s^1, y_s^1, z_s^1) of the $\mathbf{PM}_j$ (1, 2, …, m).

The requirement of resources allocations are:

1: Resource should be according to the request.

2: ∀**VM**s demand is ≤ the **PM**s' total capacity of resources.

3: ∀**VM**s each **VM** is operator by each PM according to time.

4: Assume that $\mathbf{a}_j(t)$ is group allocated to **PM**.

$\sum_{i=1}^{n}$ $\mathbf{CR}^n$ of these assigned **VM**s is ≤ the **PM**s' total resource capacity.

$$\text{For all capacity } (\forall s) = 1, \ldots, d \cdot \sum\nolimits_{\mathrm{vmi} \in \mathrm{aj(t)}} A_{i,s} \le B_{i,s} \tag{7}$$

Cost of Resources Utilization of PM

$$\mathrm{RU}_j^d = \frac{\sum_i^n = p_{ij} \times \mathrm{VM}_i^d}{\mathrm{PM}_j^d} \forall d \in \{x_s^1, y_s^1, z_s^1\} \tag{8}$$

Cost of Resources Utilization of Data Center

$$RU^{DC} = \int_{t1}^{t2} \frac{\sum_{j=1}^{m} U_j^{x_s^1} + \sum_{j=1}^{m} U_j^{y_s^1} + \sum_{j=1}^{m} U_j^{z_s^1}}{|d| \sum_{j=1}^{m} p_j^i} \partial t, \tag{9}$$

where $\mathrm{Ai} = \begin{cases} 1 \text{ if } \mathbf{CR} \text{ assigned to Ai} \\ 0 \text{ otherwise} \end{cases}$

We know every PM_j can host on any VM_i and the model for energy consumption $P_j(t)$ for PM_j's host has a linear relationship with resource utilization (as if the utlization will increase it will also effect on energy consumption [12]· The formula for achieving these parameters are given below.

Total Energy Consumption (TEC)

At time t utilization is $\mathrm{EU}(t)_j$ and energy consumption of EU(t) is depicts as P (EU (t))

$$\mathrm{TEC}_j^x = \int_{t1}^{t2} P(\mathrm{EU}(t))\, d\, t\, \mathrm{PM}_j \in P, \tag{10}$$

$$\text{Maximize} \sum\nolimits_{j=1}^{m} = \mathrm{EU}(t)_j \tag{11}$$

$$\mathrm{TEC}_j^x = \sum\nolimits_{j=1}^{m} \mathrm{EU}(t)j, \tag{12}$$

EU(t)j with j = 1, 2, …, m is the total consumed energy of the PM_j.

i ∈{1,2,…, n},j ∈{1,2,…, m}, [0;],t ∈ [0; t].

Energy Consumption of Data Center

$$\mathrm{DC}_U^{\mathrm{Energy}} = \int_{t1}^{t2} P(\mathrm{EU}(t))(x)dt + \mathrm{EU}(y) \times P_{\max} + \mathrm{EU}(z) \times P_{\max} \tag{13}$$

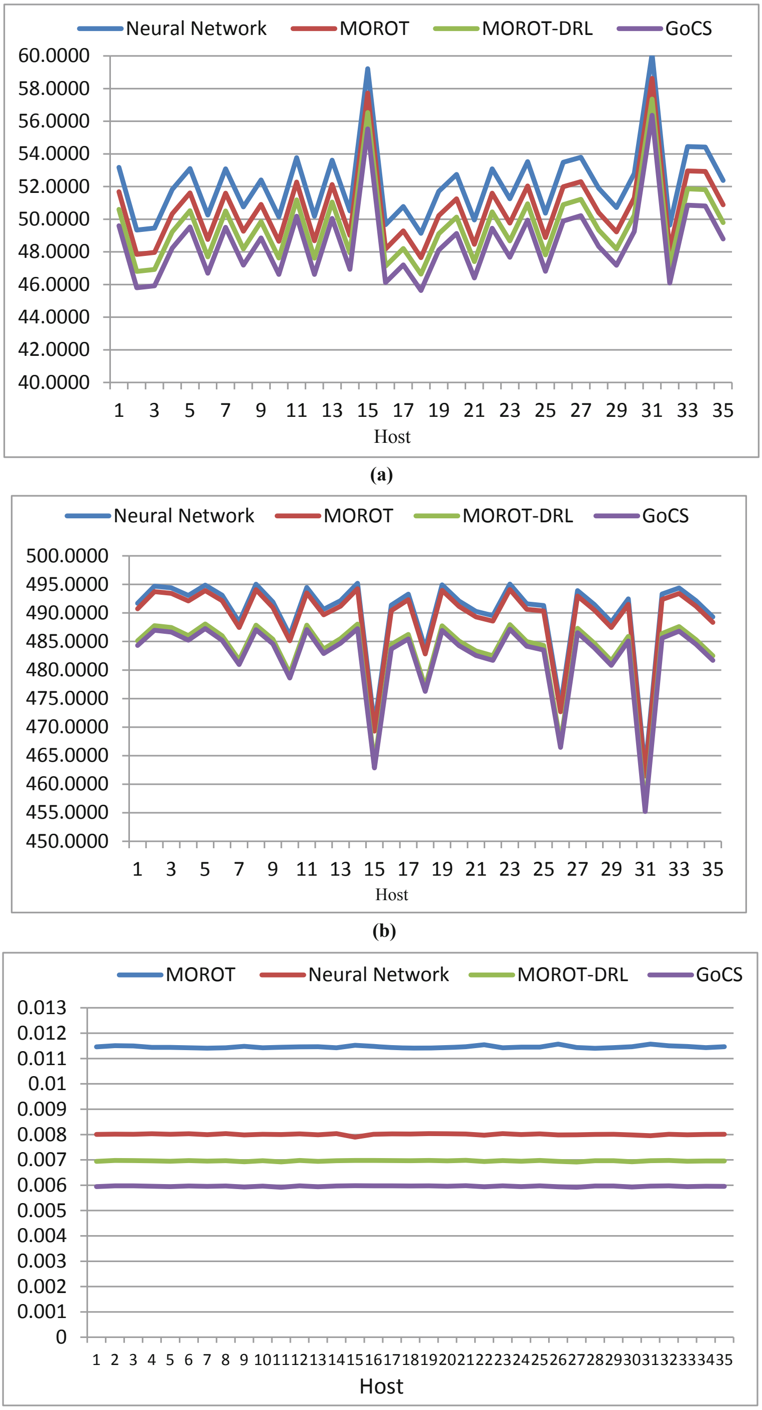

Fig. 4. (a). Comparison based on Energy consumption. (b). Comparison based on cost. (c). Comparison based on CO_2 emission

The efficiency of R^A helps to increase utilization of resources (RU) and the energy efficiency of the cloud datacenter can be generated as follows in Eq. (13).

$$\text{Maximize} = \mathrm{RU}_j^d \frac{\sum_i^n = \mathrm{p_{ij}} \times \mathrm{VM}_i^d}{\mathrm{PM}_j^d} \forall d \in \left\{x_s^1, y_s^1, z_s^1\right\} \text{s.t.Minimizes} \sum_{j=1}^{m} \mathrm{EU(t)j}, \quad (14)$$

Carbon Emissions. Total power generated is T, the power generated by the K^{th} source is S_k and ef_k is the emission factor related to K^{th} source, the total emission factor of the cloud data center can be calculated as follow.

$$\mathrm{ef} \ =_{i=1}^{k} \frac{\mathrm{Sk}}{\mathrm{T}} \mathrm{efk}$$

The proposed model MOROT-DRL is compared with the three models MOROT, neural network and one more real cloud data set. When same model is implemented in GoCS set also give more better results in energy, co2 and cost reduction.. Following Table 2 shows the Energy, cost and co2 evaluation in these entire three models (Fig. 4).

The above results show that the MOROT-DRL model performs better in the case of energy consumption, and CO_2 reduction as compared to the MOROT model and neural network and GoCS google data set in a real cloud environment. In the MOROT model resources are allocated to the user only in one cloud with the help of a minimum execution time algorithm, cyber shake seismogram, and enhanced flower pollination. The neural network is the technique of handling the incoming task in a cloud environment [37]. But the MOROT-DRL model mainly used the Q-learning technique for scheduling the incoming request in a multi-cloud environment. The q-learning handles the request while entering in the multi-cloud environment but in each cloud, the concept of cyber shake seismogram, MET, and enhanced flower pollination technique is used which efficiently works to allocate the optimized resources allocation based on the local and global method. So the proposed method gives good results as shown in graphs and tables.

5 Conclusion

Multi-cloud gives more elasticity to the users by combining multiple cloud domains and data centers. These features attract not only normal users but also the biggest companies and businesses. The requirement of cloud providers and cloud users are escalating gradually. The challenge is to handle the request and allocate the required resource. Researchers proposed many scheduling and resource allocation techniques which give good results in various parameters such as time, cost, throughput, reliability, etc. Some more are need to improve. So for this, we proposed the MOROT-DRL model implemented in real cloud environment which works on energy, cost, and CO_2 parameters. The Q-learning technique logically handles the incoming request and works as an intelligent model in a multi-cloud environment. Cyber shake seismogram workflow and minimum execution time algorithms create a queue based on minimum execution time and schedule the task in a specific cloud. The bio-inspired algorithm, i.e., enhanced flower pollination picks the task from the queue and allots the optimized resources with dynamic switching property, and local and global strategy. In the end, the comparison is made with MOROT and neural network model and GoCS, our proposed model performing superior in energy efficiency, CO_2 reduction, and cost evaluation on Gocs real data set also.

References

1. Petcu, D.: Consuming resources and services from multiple clouds. J. Grid Comput. **12**(2), 321–345 (2014)
2. Panda, S.K., Jana, P.K.: SLA-based task scheduling algorithms for the heterogeneous multi-cloud environment. J. Supercomput. **73**(6), 2730–2762 (2017)
3. Keshavarzi, A., Haghighat, A.T., Bohlouli, M.: Enhanced time-aware QoS prediction in multi-cloud: a hybrid k-medoids and lazy learning approach (QoPC). Computing **102**(4), 923–949 (2020)
4. Munteanu, V., Sandru, C., Petcu, D.: Multi-cloud resource management: cloud service interfacing. J. Cloud Comput. Adv. Syst. Appl. **3**, 1–23 (2014)
5. Ferry, N., Chauvel, F., Song, H., Rossini, A., Lushpenko, M., Solberg, A.: CloudMF: model-driven management of multi-cloud applications. ACM Trans. Internet Technol. **18**(2), 1–24 (2018)
6. Guerrero, C., Lera, I., Juiz, C.: Resource optimization of container orchestration: a case study in multi-cloud microservices-based application. J. Supercomput. **74**(7), 2956–2983 (2018)
7. Kritikos, K., et al.: Multi-cloud provisioning of business processes. J. Cloud Comput. Adv. Syst. Appl. **8**, 1–29 (2019)
8. Li, C., Zhang, J., Tang, H.: Replica-aware task scheduling and load-balanced cache placement for delay reduction in the multi-cloud environment. J. Supercomput. **75**(5), 2805–2836 (2019)
9. Mohammadi, S., Pedram, H., PourKarimi, L.: Integer linear programming-based cost optimization for scheduling scientific workflows in multi-cloud environments. J. Supercomput. **74**, 4717–4745 (2018)
10. Li, J., Lin, Y., Jia, X., Ren, K.: Multiple-replica integrity auditing schemes for cloud data storage. Concurrency Comput. Pract. Exper. **33**, 1(2019)
11. Souri, A., Rahmani, A., Rezaei, N.: A hybrid formal verification approach for QoS-aware multi-cloud service composition. Cluster Comput. **23**, 2453–2470 (2020)
12. Carvalho, J., Trinta, F., Vieira, D., Cortes, O.: Evolutionary solutions for resources management in multiple clouds: State-of-the-art and future directions. Futur. Gener. Comput. Syst. **88**, 284–296 (2018)
13. Masdari, M., Zangakani, M.: Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities. J. Supercomput. **76**, 499–535 (2019)
14. Bruno, R., Costa,F., Ferreira, P.: freeCycles - efficient multi-cloud computing platform. J. Grid Comput. **15**(1), 501–526 (2017)
15. Paraiso, F., Merle, P., Seinturier, L.: soCloud: a service-oriented component-based PaaS for managing portability, provisioning, elasticity, and high availability across multiple clouds. Computing **98**, 539–565 (2016)
16. Rashida, S., Sabaei, M., Ebadzadeh, M., Rahmani, A.: A memetic grouping genetic algorithm for cost-efficient VM placement in the multi-cloud environment. Clust. Comput. **23**(2), 797–836 (2020)
17. Khan, M.: Optimized hybrid service brokering for multi-cloud architectures. J. Supercomput. **76**, 666–687 (2020)
18. Panda, S., Gupta, I., Jana, P.: Task scheduling algorithms for multi-cloud systems: allocation-aware approach. Syst. Front. **21**, 241–259 (2019)
19. Lijin, P.: Resource allocation in multi-cloud based on usage logs. Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol. IJSRCSEIT **3** (2018)
20. Pietrabissa, A., Priscoli, F., Giorgio, A., Giuseppi, A., Panfili, M., Suraci, V.: An approximate dynamic programming approach to resource management in multi-cloud scenarios. Int. J. Control **90**, 492–503(2016)

21. Mishra, S., et al.: Energy-aware task allocation for multi-cloud networks. IEEE Access **8**, 178825–178834 (2020)
22. Carvalho, J., Vieira, D., Trinta, F.: Dynamic selecting approach for multi-cloud providers. In: Luo, M., Zhang, L.-J. (eds.) CLOUD 2018. LNCS, vol. 10967, pp. 37–51. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94295-7_3
23. Antonio, P., et al.: Resource management in multi-cloud scenarios via reinforcement learning. In: Proceedings of the 34th Chinese Control Conference, pp. 28–30 (2015)
24. Kang, S., Veeravalli, B., Aung, K.: Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi-cloud systems. J. Parallel Distrib. Comput. **113**, 1–16 (2018)
25. Chen, Z., Lin, K., Lin, B., Chen, X., Zheng, X., Rong, C.: Adaptive resource allocation and consolidation for scientific workflow scheduling in multi-cloud environments. IEEE Access **8**, 190173–190183 (2020)
26. Panda, S., Jana, P.: Efficient task scheduling algorithms for the heterogeneous multi-cloud environment. J. Supercomput. **71**, 1505–1533 (2015)
27. Farid, M., Latip, R., Hussin, M., Hamid, N.: Scheduling scientific workflow using multi-objective algorithm with fuzzy resource utilization in multi-cloud environment. IEEE Access **8**, 24309–24322 (2020)
28. Subramanian, T., Savarimuthu, N.: Application-based brokering algorithm for optimal resource provisioning in multiple heterogeneous clouds. Vietnam J. Comput. Sci. **3**(1), 57–70 (2016)
29. Thirumalaiselvan, C., Venkatachalam, V.: A strategic performance of virtual task scheduling in multi cloud environment. Cluster Comput. **22**, 9589–9597 (2019)
30. Grozev, N., Buyya, R.: Regulations and latency-aware load distribution of web applications in multi-clouds. J Supercomput. **72**, 3261–3280 (2016)
31. Zhan, W., et al.: Deep-reinforcement-learning-based offloading scheduling for vehicular edge computing. IEEE Internet Things J. **7**(6), 5449–5465 (2020)
32. Qi, Q., et al.: Knowledge-driven service offloading decision for vehicular edge computing. a deep reinforcement learning approach. IEEE Trans. Veh. Technol. **68**(5), 4192–4203 (2019)
33. Wang, Y., et al.: Multi-objective workflow scheduling with deep-Q-network-based multi-agent reinforcement learning. IEEE Access **7**, 39974–39982 (2019)
34. Baer, S., Bakakeu, J., Meyes, R., Meisen, T.: Multi-agent reinforcement learning for job shop scheduling in flexible manufacturing systems. In: 2019 Second International Conference on Artificial Intelligence for Industries, pp. 22–25 (2019)
35. Zhang, L., Wang, Q., Sun ,H., Liao, J.: Multi-task deep reinforcement learning for scalable parallel task scheduling. In: 2019 IEEE International Conference on Big Data (Big Data), pp. 2992–3001 (2019)
36. Shetty, C., Sarojadevi, H., Prabhu, S.: Machine learning approach to select optimal task scheduling algorithm in cloud. Turkish J. Comput. Math. Educ. **12**(6), 2565–2580 (2021)