A Hybrid Strategy for Reduction in Time Consumption for Cloud Datacenter Using HMBC Algorithm

ARIFULLAH ullah (✉ hi160027@siswa.uthm.edu.my)

UTHM: Universiti Tun Hussein Onn Malaysia

Tanweer Alam

Islamic Azad University

Chakir Aziza

Hassan II Mohammedia University Faculty of Science and Technology Mohammedia: Universite Hassan II Casablanca Faculte des Sciences Techniques Mohammedia

Dorsaf Sebai

UTHM: Universiti Tun Hussein Onn Malaysia

Laith Abualigah

UTHM: Universiti Tun Hussein Onn Malaysia

Research Article

Keywords: Cloud computing, Virtual machine, Task allocation, Meta-heuristic, Datacenter

Posted Date: February 23rd, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-865962/v1>

License: This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

Cloud technology is a utility where different hardware and software resources are accessed on pay-per-user ground base. Most of these resources are available in virtualized form and virtual machine (VM) is one of the main elements of visualization. In virtualization, a physical server changes into the virtual machine (VM) and acts as a physical server. Due to the large number of users sometimes the task sent by the user to cloud causes the VM to be under loaded or overloaded. This system state happens due to poor task allocation process in VM and causes the system failure or user tasks delayed. For the improvement of task allocation, several load balancing techniques are introduced in a cloud but stills the system failure occurs. Therefore we hybrid modified artificial bee colony for improvement in task allocation of VM and minimizing time consummation in cloud datacenter like makespan, total processing time, response time of algorithms, response time of datacenter and degree of imbalance. The consequences of the proposed task scheduling algorithm are matched with existing heuristic-based scheduling procedures. The experimental consequences demonstrate that our approach is efficient when it is compared with the existing algorithms and reduce 1.7 % in time consumption for cloud datacenter.

1. Introduction

Nowadays the fast growing technologies are internet of things, (IoT) fifth generation (5G) connectivity, artificial intelligence (AI) and cloud computing (CC). CC is one of the fastest growing technologies in the field of information technology which provide services to every user from anywhere at any time with the help of internet. Cloud computing has materialized as new model that distributes services at different level. Some of the main services offered by cloud provider are software as a service (SaaS), application as a service (AaaS), hardware as a service (HaaS), desktop as a service (DaaS), backup as a service (BaaS), and network as service (NaaS). These services provide by cloud according the user requirement and demands [1, 2]. These services are viable around the globe on the basic of pay as you go model therefore it is known as the avant-garde technology of the decade. One of the main characteristic of cloud computing is virtualizations where we can get any resource as virtual rather than actual. Infrastructure as a service (IaaS) is the form of CC in which different virtualizations are performed and VM is one of them best example of this model. Due to dynamic nature of VM workloads make complex situation in cloud datacenter when more user demands for task in real-time situation [3]. In the above situation if VM become underloaded or overloaded or other issues occur which affect the cloud datacenter or datacenter face issues due to VM consolidation. VM consolidation is the process of transfer VM among into physical or vice versa in order to improve the resource utilization in cloud data center [4]. Figure 1 present the VM consolidation type in cloud datacenter.

VM consolidation is important mechanism in cloud datacenter with the help of this can improve energy management and resource utilization. Different nature-inspired meta-heuristic algorithms are used to solve VM consolidation problems like VM placement, overload detection, and under of task, VM selection and energy system in VM. One of the big issues is task allocation system in VM's because during the allocation of tasks in VM, the two conditions can occur i.e. under loaded and overloaded. Both conditions

can cause the network failure or delay of response to user's request. Moreover, a proper load balancing technique is needed to overcome these problems [5]. Dynamic load balancing technique is one of the main elements or heart of datacenter in a cloud environment; because, it helps VM's to carry out the same amount of workload and deliver the service with minimal time without delay to the end user. Therefore, two well-known nature inspired algorithms were selected in improving dynamic load balancing techniques [6, 7]. So in this paper we modified artificial bee colony algorithm for improvement in VM task allocation system in data center because VM task allocation system take more time due to underloaded and overloaded situation of tasks. With the help of this modification different time section of datacenter are improved like makespan, response time of algorithm, response time of DC, total processing time and degree of imbalance. The result of proposed algorithm will be compared with other standard algorithms. The result of the paper is organized as follow. The related work about ABC algorithm and different parameter improvement are discussed briefly in Sect. 2; research contribution discuss in Sect. 3; simulation environment are briefly described in Sect. 4; performance comparison is presented in Sect. 5; statistical analysis of variance (ANOVA) significant test are discuss in Sect. 6. Section 7 discuss about conclusion and future work. Different notations are used in this paper which is mention in table 1 which helps the reader to understand the paper.

Table.1.Notation used in this paper

Symbol	Description	Symbol	Description
VM	Virtual machine	HBAC	Hybrid Bat and Artificial intelligent bee colony
LB	Load balancing	iT	Iteration Of Algorithm
CC	Cloud computing	Np	Number of requests per period
DC	Data center	Mlppt	Million instruction per second of Processing Time
PTstvm	Processing time of single tasks in VM	DC	Data center
PTtvm	Processing time of all tasks in VM	UB	Use base
CJmax	Total number of Tasks	SCR	Second cycle range
Ppn	Priority per node	SBP	Service broker policy
RP	Routing Policy	SLA	Service level agreement

2. Recent Literature

There exit several load balancing technique in literature based on single and multiple algorithms used for improvement in time consumption for cloud data center. Different parameters are used to check the

proposed technique and some of the recent literatures are given below.

Author [8], modified ABC algorithm because in the first iteration of the algorithm, the bee searched the food source randomly but in the second search step the researcher changed and improved search function that save the iteration function with the help of this changed the author improved the selection of VM method. In paper [9] the author proposed an algorithm known as osmotic bio inspired algorithm used for load balancing purpose. The author used two algorithms which were ABC and ACO algorithm for load balancing in VM where both algorithms communicated with each other with the help of osmosis theory. Due to this improvement the tasks distribution for VM improved. Author [10], proposed load balancing over slow resources (LBSR) algorithm for improvement in load balancing using the available policy rule and make change at section of the algorithm. Even though the proposed technique improved selection site but still gap exists in fitness parameter. In paper [11] the author proposed an algorithm base on ant's behavior where head node of ants move toward food source like the same rule first head node find available VM then other data move towards the VM but the data head node have no record about the VM which affect the system but the modified algorithm save the record due to that searching process improved. In [12], proposed honey bee galvanizing algorithm in which he increased honey bee forage technique with random stealing is employed for task and cargo leveling. They found out that if the VM become overload then the task moves to the neighborhood VM. If that load worth is less than threshold value, then this task was given to that VMs. The proposed technique improved the efficiency parameter as well as the average time of network. Table 1 present the summary of recent literature where it present the proposed technique name, in which section it modified along with improved parameters.

Table.2.Summary of recent literature

Technique	Parameters	Performance	Section
ACO Algorithm	Distribute the load among VM better result	Time increased	VM / load distribution, Makespan, Task
Osmotic Bio inspired algorithm	Increased make span, task number improve	Take more energy	VM / Makespan, Task number
LBSR Algorithm	Improved selection part network improve	Fitness parameter Search path need more	VM Makespan, Task number
ACO Algorithm	Distribute the load among VM better result	Time increased	VM / load distribution, Makespan, Task
HBG algorithm	Good in average speed Three parameters used	Not consistent Overall execution time not good	VM /Makespan Response time and Resource utilization

Table 2 present the summary of recent literature and Table 3 present the improvement different parameter of ABC algorithm by pervious researchers. Table 3 presents the related work about the

improvement in different parameter of ABC algorithm. This parameter modification or improvement took place just for improvement in resource allocation system of VM in cloud data center.

Table 3
Pervious improvement in ABC algorithm

References	VM section	ABC modification section	Performance parameters
[13]	VM task allocation section	initial stage	Response Time, QoS
[14]	VM rules	Depends on policies scout bee	Makespan
[15]	VM management system	initial stage	Makespan, Throughput
[16]	VM task allocation section	Global optimized section	QoS, Minimize the decision time
[17]	VM task allocation section	Search Section	Makespan, QoS
[18]	VM QOS	Initiation section	QoS, Accuracy
[19]	VM rules	Search Rule	QoS, Accuracy
[20]	VM rules	Fitness function	Makespan
[21]	VM task allocation section	Fitness function	Makespan

3. Research Contribution

This section highlights the contribution made by the author's in this paper. The research mainly aims to improve the task allocation system of VM's in cloud data center by enhancing load balancing approach. Due to the modification in tasks allocation system different parameter of time consumptions improved in cloud datacenter. Figure 2 presents the frame work of the proposed approach. Our contribution of the paper can be summarized as below.

[1] A survey of existing improvement in different section of ABC algorithm used for load balancing purpose.

[2] The proposed algorithm modification sections are discussed with explanation.

[3] The proposed algorithm addresses the VM violation issue like under/overloaded task distribution in cloud datacenter.

[4] The proposed algorithm improved different time consumptions parameters in cloud data center.

[5] The proposed algorithm results are comparing with well-known algorithm.

[6] Discuss the result along with future implementation of the proposed algorithm.

3.1 Improved Employed bee in the proposed HMBC Algorithm

The proposed HMBC Algorithm inherits the basic idea from ABC algorithm and tries to implements load balancing technique for improvements in task allocation system of VM in the cloud computing. Different parameters like makespan, total processing time, response time of algorithms, response time of datacenter and degree of imbalance are used to check the performance of the proposed algorithm in the cloud computing. For that reason, three main sections of ABC algorithm are modified in the proposed HMBC i.e. onlooker bees, employed bees and load balancer. Normally, an ABC algorithm consists of different parameters which are employed bees, onlooker bees, and scout bees. During the execution process, they change into sub-groups. Initialization of the population consist of two groups of bees which are known as employed and onlooker both containing the same number of bees. When we analyzed the algorithm, employed bee and onlooker bee used same equation for obtaining the candidate solution which makes an imbalanced choice or worst case can occur due to the previous update rule [22]. Therefore, we used new update rules in this section which we derived from Bat algorithm update rule. Bat algorithm use virtual Bat for the update rules for their position or food source as given in the Eqs. 1 and 2.

$$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*)f_i$$

1

$$X_i^t = X_i^{t-1} - V_i^t$$

2

Where V_i^t present the velocity and X_i^t present the location at t (cycle) iteration in space or possible dimension solution [23, 24]. Table 4 shows the Pseudocode of improved employed bee in the proposed algorithm.

Table .4.Improved Employed bee in the proposed Algorithm

Algorithm 2: Employed bee section	
1	For i = 1 to n
2	Sen the population of bees in VM at random search method.
3	$V_i^t = V_i^{t-1} + (X_i^{t-1} - X_*)f_i$ $X_i^t = X_i^{t-1} - V_i^t$
4	Calculate the fitness value of each VM $f_j = (p_j \times m_j bw_j$ Where j = index of VM that ith bees found.
5	End for.

3.2 Proposed Algorithm selection approach for Onlooker bee

When the employed bee of ABC algorithm exploits the nectar source and shares all the information upon returning to beehive about the food source by making wangle dance. As mentioned in the Eq. (3) with the roulette wheel selection the onlooker bee gets the information with probability and the fitness value of the solution (quality of food source). Onlooker bee try better food sources as per rules according to the current cycle. After the first cycle, the successive cycles start with employed bee moving to the food source position update by onlooker bee. During this process, some of the food source remains unchanged for predetermined number of trails which causes time consumption in DC's. Therefore, we improve the search selection approach for onlooker bee by taking range sector along with the fitness value [25]. So, the average fitness value obtained from the employed bee is calculated as.

$$Avg_t^{pit} \frac{I}{n} \sum_{i=0}^n fit_i$$

3
Table .5. Selection approach for onlooker bee in the proposed Algorithm

Algorithm 2: Onlooker bee phase	
1	The first (m) onlooker bees as food source (VM) are choice with the highest fitness value and performed neighborhood search.
2	Numbers of employed bee are sent in to new position for food source and when they return the share the information.
3	$Avg_t^{pit} \frac{1}{n} \sum_{i=0}^n fit_i$
4	The onlooker bee calculate the fitness value of food source according eq in line 5.
5	$fit_{ij} = \frac{\sum_{i=1}^n data_{length_{i,j}}}{datasize}$
6	The onlooker bees choose the best food source (VM) and assign task to VM.

3.3 Load balancer section in the proposed HMBC Algorithm

After the completion of all iterations, task is assigned to different VM's using load balancer section. Based on the standard deviation, it checks the processing time of a single VM or a group of VM's. Here, a VM is also checked for overloaded/under loaded. It is calculated by using the standard deviation which is given below.

$$SD = \sqrt{\frac{1}{n} \sum_{j=0}^n (X_j - X)^2}$$

4
Processing time of data by a single VM can be measured by;

$$X_j = \sum_{i=1}^d \frac{data-length(i)}{Capacity}$$

5
Means of processing time of all VMs can be calculated as;

$$X_j = \sum_{i=1}^d \frac{data-length(i)}{n-1}$$

6
In the given section load balancer section, standard deviation is used for the distribution of data to a single or a group of VM's in a cloud data center [26]. Figure 2 shows the working of proposed algorithm.

4. Simulation Environment

The simulations consist of different experiments which consist of different data centers, different number of VM's running on the physical host and different numbers of tasks are given. Table 3 show the honey bee presentation in VM implementation for cloud datacenter.

Table 6
Honey bee cloud computing environment

Bees	Cloudlets
Source of food	Mapping of cloudlet to VM
Bee getting worn-out at the food source	VM is overloaded
Bee searching the new food source	Sending the task from overloaded virtual machine to another underloaded virtual VM

Table 7 shows the system parameters settings. In the simulation steps, Round-robin, Bat, Artificial bee colony, Genetic, Ant colony optimization algorithms are compared with the proposed algorithm in-terms of makespan, total processing time, response time of algorithms, response time of datacenter and degree of imbalance by using different number of tasks. CloudSim used for simulation process, Table 9 shows the parameters some of parameters are change at each experiment.

Table 7
Network parameter setting

Type	Parameter	Value	Type	Parameter	Value
Region	From 1 to 4	5	Proposed algorithm	Maximum number of Iteration	100
Data center	Number of data center	5		Number of Honeybees	25
	Number of hosts	100/1000		Number of scout bee	20
	Type of Manager	Time and space		Number of Onlooker bee	5
	Bandwidth	1000	Dataset	Data center	No/VMs
Virtual machine	Total number of VM	30/20/20	SWF		
	Number of processor per virtual machine	4	Size 400K/200k	D1 ,D2,D3	D1 = 5
	Virtual machine memory	512	PC	D4,D5	D*5 = 30/50
	Bandwidth in bit	1000		64-bit version	12 GBRM
	VM image size	1000	Task/Data	Number of task	100/1000
Memory	Total memory	204800Mb		Length of task	100/200/400 byte
	Number of processors	4 per VM		Number of processor per requirement	250 KB
	Total processor	120	Cloudlets	Total number of task	400/600
	Storage Memory	100000Mb	CPU	MIPS OF PE/Number of PFs VM	2500/1
	Viable memory	10000	Proposed algorithm	Maximum number of Iteration	100
No/VMs	DC/Policy	12 GBRM		Random	Windows 2010

Table 4 present the different parameters some of these parameters are change darning each experiments like task number, VM number and data size.

5. Performance Comparison

In this section we present all the result of different parameters which are used to measure the performance of the proposed algorithm like makespan, total processing time, response time of algorithms, response time of datacenter and degree of imbalance. These parameters are checked with different simulation steps their result are explain below.

5.1 Performance based on makespan

In this sub-section the performance of proposed algorithm measured based on makespan time. For the paper makespan measured as total time taken by resource to complete the executing of all tasks in a cloud data center. According to [27], Makespan in a cloud datacenter is given time to simulation or a complete time of simulation. Normally, different research presents makespan as total execution time of data in a datacenter [28]. Here the analysis has been done taking 7 different scenarios taking 200 to 1600 tasks and the makespan time is calculated from the areas.

$$Makespan = \frac{Giventimetothetnetwork}{Complitetimeofnetwork} * 100$$

Here five different algorithm like round-robin, Bat, artificial bee colony, genetic algorithm, ant colony optimization algorithms and proposed algorithm are evaluated by different number of tasks from 200 to 1600. The result of proposed algorithm makespan are, 187/200,293/400,338/600,396/800,443/1000,478/1200 and 507 /1600 which are the most less number of time taken by any algorithm for completing number of takes in the experiments. Figure 4 to Fig. 7 present the makespan result and Table 8 presents the overall result of all algorithms.

Table .8.Result analysis of different workflows for makespan time

Makespan						
Task Number	RR	BAT	ABC	GA	ACO	HMBC
200	200	197	190	195	190	187
400	300	282	284	287	288	293
600	350	322	328	328	330	338
800	400	389	387	390	391	396
1000	450	420	433	436	439	443
1200	480	470	472	474	476	478
1600	510	490	500	505	504	507

5.2 Performance based on processing time

In this sub section the evaluation performance of the proposed algorithm is calculated based on processing time and the result are measured with other algorithms. The equation of total processing time is given as: $Processing\ Time = Cloudlet_{length} / VM_{Mips} \times PE_{Number}$. These processing times used for task are offloaded section only. Based on table 9 the result of proposed algorithm are 9956.78, 1090.89, 1134.78, 1156.89, 18.76.90 and 1811.89 which are the most lowest total processing time in the simulation Table 9 and Fig. 8 present the overall result.

Table .9. Total processing time

Total processing time						
Task Number	RR	BAT	ABC	GA	ACO	HMBC
200	1300.89	1245.67	1223.67	1156.78	1089.56	9956.78
400	1500.67	1478.89	1389.89	1245.78	1123.67	1090.89
600	1765.65	1645.89	1567.78	1356.89	1234.78	1134.78
800	1800.78	1769.89	1543.67	1456.89	1289.67	1156.89
1000	2234.89	2145.89	2089.89	1987.89	1876.78	1756.89
1200	2489.89	2356.78	2245.78	2167.98	2078.89	18.76.90
1600	2578.89	2345.78	2234.78	2198.09	1878.89	1811.89

5.3 Performance based on response time of algorithms

In this sub section the evaluation performance of the proposed algorithm is calculated based on response time of the algorithm the result are measured with other algorithms. Based on table 10 the results of proposed algorithm are 76.89, 158.78, 189.90, 250.90, 310.90 and 360.90 these are the most fast response time of algorithm these result are conduct from the proposed Algorithm.It means the proposed algorithm provide better result as compare to other algorithm Table 10 and Fig. 9 present the overall results.

Table .10. Response time of algorithms

Response time of algorithms						
Task Number	RR	BAT	ABC	GA	ACO	HMBC
200	109.89	107.67	105.67	99.89	89.89	76.89
400	189.89	178.90	176.89	170.809	167.89	158.78
600	234.78	209.90	202.89	198.89	190.90	189.90
800	302.89	289.90	278.90	267.89	260.89	250.90
1000	350.89	340.90	330.90	310.90	300.90	289.09
1200	378.89	365.89	350.89	345.90	330.90	310.90
1600	412.78	400.89	390.89	378.90	370.89	360.90

5.4 Performance based on response time of datacenter

In this sub section the evaluation performance of the proposed algorithm is calculated based on response time of datacenter the result are measured with other algorithms. Based on table 11 the results of proposed algorithm are 81.90, 100.67, 165.90, 169.89, 250.90 and 265.90 these are the fastest response of algorithm to the datacenter and these are the result of proposed algorithm. Table 11 and Fig. 10 present the overall result.

Table .11. Response time of datacenter

Response time of datacenter						
Task Number	RR	BAT	ABC	GA	ACO	HMBC
200	97.90	93.78	90.78	88.90	86.90	81.90
400	140.89	130.90	120.89	118.90	113.90	100.67
600	200.89	189.90	180.89	178.89	179.90	165.90
800	240.89	210.90	192.90	186.90	190.90	169.89
1000	280.90	270.90	250.90	240.90	220.90	200.92
1200	300.90	287.90	278.90	266.89	260.90	250.90
1600	320.90	300.89	290.90	280.90	270.90	265.90

5.5 Performance based on degree of imbalance

In this sub-section the proposed algorithm is tested based on the degree of imbalance and compared with other present algorithms the given equation presents degree of imbalance.

$\{D_i\} = (\{T_{\text{max}}\} - \{T_{\text{min}}\}) / \{T_{\text{avg}}\}$. Where $\{T_{\text{max}}\}$ and $\{T_{\text{min}}\}$ are minimum and maximum $\{T_i\}$ present the all VM's $\{T_{\text{avg}}\}$ is present the average $\{T_i\}$ VM. Based on table 12 the proposed algorithm provide most efficient result as compare to the standard algorithm Table 12 and Fig. 11 present the overall result.

Table .12.Result analysis of different workflows for degree of imbalance

Degree of imbalance						
Task Number	RR	BAT	ABC	GA	ACO	HMBC
200	60.89	56.90	47.45	45.34	43.78	40.89
400	90.89	89.90	87.67	80.34	77.45	70.34
600	120.90	110.56	100.89	89.78	86.67	80.56
800	180.90	170.67	167.67	162.67	160.56	156.45
100	200.90	198.90	190.45	188.67	180.67	176.78
1200	220.90	210.67	200.78	190.56	189.67	180.67
1600	250.78	245.67	240.67	230.56	220.67	200.78

6. Statistical Analysis Of Variance (Anova) Significant Test

The one-way analysis of variance (ANOVA) is used to determine whether there are any statistically significant differences between the means of two or more independent (groups). One-way ANOVA compares the means between the groups and determines whether any of those means are statistically significantly different from each other and it used null hypothesis as shown in Equation

$H_0 : \mu_1 = \mu_2 = \mu_3 = \dots = \mu_k$: Where μ present group means and k =number of groups. In one-way ANOVA there are two possible hypotheses which are null hypothesis (H_0) and alternative hypothesis (H_1). Null hypothesis (H_0) means there is no difference between groups and equality between means whereas alternative hypothesis (H_1) presents that there is difference between the groups or means. If, one-way ANOVA returns a statistically result that can be accepted as alternative hypothesis (H_1), which indicate that there are at least two group means that are statistically significantly different from each other. The performance of the proposed algorithm is evaluated based makespan, total processing time, response time of algorithms, response time of datacenter and degree of imbalance. The results of different parameters consist of different tasks number from 200/2000. The use of ANOVA had produced the collection of all results that are more precise with single result. Furthermore, origin software is used for ANOVA testing after all the simulation experiments were performed and their results are put in origin software for the result the results are put in Fig.12 and Fig.13.

After applying the results from the Table 4.6 up to Table 12 and Fig. 7 upto Fig. 13 the different result of time section are used based on the ANOVA test result the proposed algorithm improve 1.7% in time reduction for cloud datacenter. It means the proposed algorithm has greatly improved the load balancing/task allocation in VMs for cloud computing for time consumption of datacenters. The above simulations are performed based on non-preemptive load balancing approach for cloud datacenters.

7. Conclusion And Future Work

VM tasks distributions is still one of the advancing exploration problems in cloud datacenter. In this paper for improvement in VM data distribution load balancing approach used where we modified ABC algorithm which was on the base of honey bee foraging existing heuristic algorithm. The proposed algorithm aims to minimize the overall response time of data center (i-e) makespan, processing time, response time of datacenter ,response time of algorithms and degree of imbalance. It distributed the work load between different VM with the rule of availability of VM.It manage the overall time according to the threshold time. The simulation result shows that purposed algorithm improve in time consumption of datacenter for cloud computing. The proposed load balancing algorithm shows better performance for a large number of assigned tasks and data size as compare to the other standard algorithms. In upcoming paper we implement the proposed algorithm for predication of host in cloud datacenters.

Declarations

Data Availability Statements

No specific Data are used because it is review papers all those paper which are study and used in this paper are cited in the paper.

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.

Author Contribution

All authors are equality contribution for making paper.

References

1. Jena, T., & Mohanty, J. R. (2018). GA-based customer-conscious resource allocation and task scheduling in multi-cloud computing. *Arabian Journal for Science and Engineering*, 43(8), 4115–4130
2. Rangwani, D., & Om, H. (2021). A Secure User Authentication Protocol Based on ECC for Cloud Computing Environment. *Arabian Journal for Science and Engineering*, 46(4), 3865–3888

3. Adhikari, M., & Koley, S. (2018). Cloud computing: a multi-workflow scheduling algorithm with dynamic reusability. *Arabian Journal for Science and Engineering*, 43(2), 645–660
4. Alkatheiri, M. S. (2021). PCOS—Privacy-Controlled Offloading Scheme for Secure Service Data Offloading in Edge-Internet of Things-Cloud Scenario. *Arabian Journal for Science and Engineering*, 1–14.
5. Ouhamme, S., Hadi, Y., & Ullah, A. (2021). An efficient forecasting approach for resource utilization in cloud data center using CNN-LSTM model. *Neural Computing and Applications*, 1–13.
6. Ullah, A. (2019). Artificial bee colony algorithm used for load balancing in cloud computing. *IAES International Journal of Artificial Intelligence*, 8(2), 156
7. Ullah, A., & Nawi, N. M. (2020). Enhancing the dynamic load balancing technique for cloud computing using HBATAABC algorithm. *International Journal of Modeling, Simulation, and Scientific Computing*, 11(05), 2050041
8. Ullah, A., Nawi, N. M., & Khan, M. H. (2020). BAT algorithm used for load balancing purpose in cloud computing: an overview. *International Journal of High Performance Computing and Networking*, 16(1), 43–54
9. Ouhamme, S., Hadi, Y., & Arifullah, A. (2020). A hybrid grey wolf optimizer and artificial bee colony algorithm used for improvement in resource allocation system for cloud technology.
10. Gamal, M., Rizk, R., Mahdi, H., & Elnaghi, B. E. (2019). Osmotic bio-inspired load balancing algorithm in cloud computing. *IEEE Access*, 7, 42735–42744
11. Gamal, M., Rizk, R., Mahdi, H., & Elhady, B. (2017, September). Bio-inspired load balancing algorithm in cloud computing. In *International Conference on Advanced Intelligent Systems and Informatics* (pp. 579–589). Springer, Cham.
12. Shen, L., Li, J., Wu, Y., Tang, Z., & Wang, Y. (2019, May). Optimization of artificial bee colony algorithm based load balancing in smart grid cloud. In *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)* (pp. 1131–1134). IEEE
13. Kruekaew, B., & Kimpan, W. (2014, March). Virtual machine scheduling management on cloud computing using artificial bee colony. In *Proceedings of the International MultiConference of engineers and computer scientists* (Vol. 1, pp. 12–14).
14. Ajayi, O. O. (2017). *A Class-based Virtual Machine Consolidation for Improved Quality of Service and Energy Conservation in Cloud Computing* (Doctoral dissertation).
15. Tawfik, M. A., Bahgat, A., Keshk, A., & Torkey, F. A. (2015). Artificial Bee Colony Algorithm for Cloud Task Scheduling. *IJCI. International Journal of Computers and Information*, 4(1), 1–10
16. Babu, K. R., & Samuel, P. (2016). Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud. In *Innovations in bio-inspired computing and applications* (pp. 67–78). Cham: Springer
17. Sethi, N., Singh, S., & Singh, G. (2019). Improved Mutation-Based Particle Swarm Optimization for Load Balancing in Cloud Data Centers. In *Harmony Search and Nature Inspired Optimization Algorithms* (pp. 939–947). Singapore: Springer

18. Saif, M. A. N., Niranjan, S. K., & Al-Ariki, H. D. E. (2021). *Efficient autonomic and elastic resource management techniques in cloud environment: taxonomy and analysis* (pp. 1–38). Wireless Networks
19. Janakiraman, S., & Priya, M. D. (2021). Improved Artificial Bee Colony Using Monarchy Butterfly Optimization Algorithm for Load Balancing (IABC-MBOA-LB) in Cloud Environments. *Journal of Network and Systems Management*, 29(4), 1–38
20. Alatawi, H. S., & Sharaf, S. A. (2021). Hybrid Load Balancing Approach based on the Integration of QoS and Power Consumption in Cloud Computing. *International Journal*, 10(2).
21. Negi, S., Rauthan, M. M. S., Vaisla, K. S., & Panwar, N. (2021). CMODLB: an efficient load balancing approach in cloud computing environment. *The Journal of Supercomputing*, 1–53.
22. He, Z., Ma, C., Wang, X., Li, L., Wang, Y., Zhao, Y., & Guo, H. (2014). "A modified artificial bee colony algorithm based on search space division and disruptive selection strategy." *Mathematical problems in engineering* 2014
23. Alsalibi, B., Abualigah, L., & Khader, A. T. (2021). A novel bat algorithm with dynamic membrane structure for optimization problems. *Applied Intelligence*, 51(4), 1992–2017
24. Sefati, S., Mousavinasab, M., & Farkhady, R. Z. (2021). Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation. *The Journal of Supercomputing*, 1–25.]
25. Jyoti, A., & Shrimali, M. (2020). Dynamic provisioning of resources based on load balancing and service broker policy in cloud computing. *Cluster Computing*, 23(1), 377–377
26. Kumar, M., Sharma, S. C., Goel, A., & Singh, S. P. (2019). A comprehensive survey for scheduling techniques in cloud computing. *Journal of Network and Computer Applications*, 143, 1–33
27. Miao, Z., Yong, P., Mei, Y., Quanjun, Y., & Xu, X. (2021). A discrete PSO-based static load balancing algorithm for distributed simulations in a cloud environment. *Future Generation Computer Systems*, 115, 497–516

Figures

Figure 1

Type of VM consolidation

Figure 2

Proposed algorithm

Proposed Algorithm Pseudo code

```
1  For i=1 to n
2  Sen the population of bees in VM at random search method.
3  Calculate the fitness value of each VM  $f_j = (p_j \times m_j bw_j)$ .
   Where  $j$  =index of VM that ith bees found.
4  End for.
5   $fit_{ij} = \frac{\sum_{i=1}^n data\ length\ i,j}{Evaluate\ the\ capacity\ of\ VM}$ 
6  Update the fitness value
7  End for
8   $fit_{ij} = \frac{\sum_{i=1}^n data\ length\ i,j)2}{data\ size}$ 
9  The onlooker bees choose the best food source (VM) and assign task to VM.
10 If food source =Null then.
11 Send previously scout bee on to the system (VM) by randomly.
12  $C_j = P_{ej \times M_{ij} \times B_{wj}}$ .
13 End if.
14  $SD = \sqrt{\frac{1}{n} \sum_{j=0}^n (X_j - X)}$ 
15 Processing time of data in single VM can be measured as.
    $X_j = \sum_1^d \frac{data - length(i)}{Capacity}$ 
16 Processing time of data in group of VM can be measured as.


---


    $X_j = \sum_1^d \frac{data - length(i)}{n - 1}$ 
17 If equation in line 15 and 16  $\leq$  (Average processing time of VM) it means the system balanced.
18 If equation line 15 and 16  $>$  (Average processing time of VM) it means the system not balanced.
19 End if
20 End for
```

Figure 3

Pseudo code proposed algorithm

Figure 4

Makespan of 400 tasks

Figure 5

Makespan of 600 tasks

Figure 6

Makespan of 800 tasks

Figure 7

Total makespan

Figure 8

Total processing time comparison

Figure 9

Total response time comparison

Figure 10

Total response time of datacenter comparison

Figure 11

Degree of imbalance comparison

Figure 12

ANOVA result

Figure 13

ANOVA result