Self-Improved Black Widow Algorithm for Cloud Load Balancing

[1*]Koppula Geeta and [2]V. Kamakshi Prasad

Assistant Professor,Department of Computer science and engineering,Rajiv Gandhi University of Knowledge Technologies, Basar, India.

[1*]geetakoppula6@gmail.com

[2]Senior Professor,Department of Computer Science and Engineering
JNTUH College of Engineering, Hyderabad, India.

Abstract. Recently, cloud computing has emerged as a predominant technique in the realm of information technology. The increasing demand of cloud computing platform in terms of computing and data storage so that the cloud providers offers numerous data centers present at various locations across the world. These data centers comprise of millions of IT servers. Due to the fact, data centers consume excess energy and liberate high amount of carbon footprint. This lead to create dangerous effect on the environment. In this paper, a self improved black widow optimization algorithm (SI-BWO) based load balancing incloud computing. While receiving a task request from the client for accessing task, the PM assign several VM on the basis of multi-objective such as power consumption, migration cost, memory usage, and load balancing. The parameters like response time, Turn around time and server load are consider for load balancing. To attain better load balancing, the suitable VM is selected by employing proposed optimization. Finally, the proposed SI-BWO model has been computed based on make span, memory usage, migration cost, power consumption, response time, server load, throughput, turnaround time, and convergence.

Key words: cloud computing, data center, load balancing, power consumption, memory usage

1 Introduction

Due to the increasing growth of information technology, large amount of user data and applications needs more computational processing and storage location for these data. These data are stored in various data centers present in various landmarks across the globe. At present, various online media like Face book, Twitter, Amazon etc [3][5]. has their own cloud for storing data. Cloud storage is a version of internet storage technique where number of complex files, managing tools, applications and user data are present. Subsequently, many number of client request are accessed in these cloud storage. Each cloud storage requires minimum one server for storage of data integrated

K. R. Madhavi et al. (eds.), *Proceedings of the International Conference on Computational Innovations and Emerging Trends (ICCIET 2024)*, Advances in Computer Science Research 112,
https://doi.org/10.2991/978-94-6463-471-6_101

with the web. It stores the copy of file, which has been sent to the cloud storage by the client.

Cloud computing is primarily share the data through online. Virtualization techniques is employed in cloud computing, which accessing numerous IT sources available in various locations with the help of unified virtual resource pool. Cloud computing provides respective result based on the request of client [1]. The delivering of cloud service is classified into three types. They are SaaS, IaaS, and PaaS. Paas model offers framework for programme advancement. SaaS model offers software and services, IaaS model gives sophisticated environment. The cloud data centers has the ability to access various apps across the globe [6].

In cloud computing, load balancing is the crucial and essential work for allocating the task in the VM present at the data storage. It enhancing the system performance by preventing bottleneck caused by non uniform distribution of load. The efficient load balancing minimizes the response time of client request and improves the system performance. Even though, error occurs due to the failure of components, the cloud computing provides continuous service to the system. It increases the lifespan of structure by minimizing strain on the hardware components. In this cloud server accept numerous client request and few servers are under processing this leads to non-uniform sharing of loads, which affects the system performance.

The conventional cloud service approach is difficult to fulfill the increasing demand of location for data storage and numerous computational performance [4]. The transmission of huge data around the globally present data centers increases transmission cost. Every cloud needs scheduling order, to estimate execution sequence. The power consumption of data centers increasing by 12 percentage [2]. Every cloud has different scheduling algorithm. The scheduling of task involves response time, energy utilization and the ability to balancing loads. The load balancing algorithms like CDLB and DDLB regarding the service rate, queue length, and the target time of client request. This paper proposes a new cloud load balancing strategy with the following contribution.

- Proposing a new SI-BWO model for balancing the load in the cloud environment under various constraints.

This paper organization is as follows: Section 2 addresses the literature work of load balancing in cloud. Section 3 discusses about Formulated System model and derived multi-objective function, section 4 describes about SI-BWO model based load balancing in cloud environment, section 5 describes the result and discussion of the proposed model

2 Literature Review

In 2023, K. DhanaSree Devi et al.[1] has developed DLB to store IoT for load balancing in cloud. In this approach, DLB has three procedures namely balancing the managing parameters of the cloud, assigning the position and achieved minimal delay. It supported huge data and various parameters are calculated like RT, MS, AO and MT. This method when compared with conventional approaches like TA, ESCE, and ESCE+TA gave outstanding outcome for load balancing.

In 2023, Imane Aly Saroit et al.[2] has introduced LBCC-Hung method to overcome the challenge of load balancing in VM. The result of this approach was evaluated and compared with MIN-MIN and FCFS approaches. LBCC-Hung outperformed in MS and throughput and VM utilization deviation.

In 2023, J.RobertAdaikalaraj et al.[3] has proposed an ILO with Min-Max framework for enabling VNE for genetic operators parallelization. The proposed approach was outperformed in load balancing and power conservation in terms of minimal processing time cost. The applicability of data centers was greater and reached approximately 80 percent, energy consumed by data center was lowered by 49.13 percent and huge transmission of VM was lowered by 94.5 percent.

In 2023, Ajay Jangra et al. [4] has introduced load balancing model for efficient re allotting of resources about healthcare cloud environment. They used certain frameworks for resource rescheduling namely GA, SARSA, and Q-learning. This approach required less MS and consumption of low latency time for estimating solution in load balancing health care environments. This approach was simulated in MATLAB. This method achieved high throughput value and lower MS value compared with existing methods.

In 2022, Mohammad Haris et al. [5] has proposed MMHHO for dynamic load balancing. Updation of HHO search space by employing the MRFO framework regarding cost, RT, and utilization time. By modifying the wait time of the task, it increased the system performance in terms of improving throughput of VM, load balancing between VM, and balancing between preferences. This approach was executed in CloudSim tool. The proposed MMHHO load balancing algorithm well performed when compared with other algorithms.

3 Formulated System model and SI-BWO model based load balancing in Cloud Environment

A. Problem statement with solution model

Various parameters namely transmission cost, memory utilization, consumption of power and load balancing factors like Turnaround time, server load, response time are considered during the designing of system model. Cloud computing differ by its extensive operation, also it offers different kinds of services based on the request of cloud client, it gets influenced by economies of scales and offers highly demand assistance via virtualization when compares with the designing of conventional decentralized network. The proposed model consists of numerous data centers termed as PM, in which numerous VM present in every data centers process the client request. Assuming a cloud with numerous PM,which has only a less number of physically diverse hosts.

Every host is identified by its identification number, list of processing data, bandwidth, memory storage, processing speed in terms of MIPS and other parameters.

Every PM consists of N number of VMs and every host consists of numerous VMs. The host and VMs has similar features. Let $Cl = \{P^{M1}, P^{M2}, \cdots, P^{Mn}\}$, where Cl points out the cloud and P^{M1}, P^{Mn} points out the 1^{st} and n^{th} PMs present in the cloud. The PMs are points out as $P^{Mn} = \{V^{M1}, V^{M2}, \cdots, V^{MN}\}$, where V^{M1} and V^{MN} indicates the 1^{st} and N^{th} VM. Every VM has two levels. They are active and idle levels. The VM consumes 60 percent of energy in idle level when compared with VM in active level. The requests are collected from various clients and transmit it to the Central load balancer or sequence scheduler for resource mapping. In VM, every cloud consumer has various jobs to perform but every VM is responsible to perform each request at a time. Let us consider number of cloud consumers be j and each has k number of tasks to perform. The request of the client is denoted as $U_j = \{T^1, T^2, \cdots, T^k\}$. With the increasing client input task regarding various sources lead to the need of load balancing. The cloud system accepts $n0$ numerous input, it can expressed as $(T^1, T^2, \cdots, T^{n0})$. The VM managing tool accept input job from the job queue and acquires detail about the about active VM, relevant sources about the job present in the host and the length of local task queue present in the host.

B. Derived Multi-objective functions

The essential parameters of multi-objective functions such as power consumption,migration cost, and memory utilization. The efficient load balancing can be achieved by considering response time, turn- around-time and server load. The VM in terms of lower multi-objective condition is suitable for best VM for performing the job in task queue. The optimal finding objective of VM is shown in Eq.(1)

$$Of = \min(W^1 * P_c + W^2 * M_c + \\ + W^3 * M_u + W^4 * R_T + W^5 * TAT + W^6 * S_L\} \tag{1}$$

Where $W^1 - W^6$ indicates the calculation of weight function using chaotic-chebychev map. Chebychev map, $c^{k+1} = \cos(0.5 * \cos^{-1}(c^k))$. Additionally, P_c indicates power consumption, M_c indicates migration cost, M_u indicates memory utilization, R_T indicates response time, TAT indicates turn-around time, and S_L indicates server time.

Power Consumption:

It is the major factor for load balancing process. The accurate ED of all dynamic PM at a time evaluate the power consumption of the system. The minimal ED improves the performance of load balancing. When a PM cannot operate, allocation is not possible in the respective PM. The expression to evaluate PEF of each active node is shown in Eq.(2) here, $P^{hc} = 0.1$ and $P_{c\max} = 1$. Additionally, μ^{hc} indicates CPU utilization.

$$P_c = \frac{1}{n * N}\left[\sum_{m=1}^{n}\sum_{M=1}^{N} P^{hc}.P_{c\max} + (1 - P^{hc}).\mu^{hc}.P_{c\max}\right] \tag{2}$$

$$\mu^{hc} = \frac{1}{3}\frac{Q_c^{CPU}}{A_{nc}^{CPU}} + \frac{Q_c^{memory}}{A_{nc}^{memory}} + \frac{Q_c^{BW}}{A_{nc}^{BW}} \tag{3}$$

Here n and N indicated the counting of PM and VM correspondingly. Q_c^{CPU}, Q_c^{memory}, Q_c^{BW} indicated CPU utilization, memory utilization and bandwidth utilization respectively, A_{nc}^{CPU}, A_{nc}^{memory} ,and A_{nc}^{BW} indicated total CPU assessment in PM, total availability of CPU in PM, total bandwidth accessible in PM.

Migration cost:

The increasing number of transmissions increase the migration cost of VM. The optimal load balancing system must have less number of motion. The expression of migration cost for arranging the entire cloud is shown in Eq.(4)

$$M_c = \frac{1}{n * N}\left[\sum_{m=1}^{n}\sum_{M=1}^{N}\left(\frac{c}{d}\right)\mu^{hc}\right] \tag{4}$$

Memory Utilization:

Memory utilization means organizing and effectively utilizing the available memory within the system. Heap structure is a crucial way of utilizing the location in memory and is developed based on the benefits for VM. The memory utilization for the entire cloud is evaluated in Eq. (5)

$$M_c = \frac{1}{n * N}\left[\sum_{m=1}^{n}\sum_{M=1}^{N} \frac{1}{2}\left(\frac{Q_c^{CPU}}{A_{nc}^{CPU}} + \frac{Q_c^{memory}}{A_{nc}^{memory}}\right)\right] \tag{5}$$

Response time:

The mean response time is calculated based on Eq.(6) In this i refers to task index, d_i refers to distance between task i input time and the response time of initial system task.

$$T_R = \frac{\sum_{i=1}^{I} d_i}{N} \tag{6}$$

Turn-around time:

The evaluation of turn-around time for load balancing is expressed in Eq.(7), In this $d.w^a t_i$ refers to the task waiting time of i resource, le_i refers to length of the task and $\sum_{p=1}^{M}(t.R_i)^p$ points out the quantity of resources p taken for task i.

$$TAT = \sum_{i=1}^{I}\left(\frac{d.w^a t_i + le_i}{\sum_{p=1}^{M}(t.R_i)^p}\right) \tag{7}$$

Server load:

The evaluation of mean load of the server is expressed in Eq. (8)

$$T_L = \sum_{j=1}^{M} v_j(t_j - t_{j-1}) \tag{8}$$

4 SI-BWO: Mathematical Model

In order to achieve effective load balancing, this paper proposes a self improved black widow optimization in cloud environment. Swarm intelligence optimization is a type of metaheuristic optimization which attain high optimization performance. Similarly, the black widow algorithm finds best solution based on the attributes of initial population, crossover, cannibalism and mutation. This type of algorithm provides high convergence speed and optimization accuracy. There are four steps in this optimization. They are initial population, procreate, cannibalism, mutation.

4.1 Initial population

While addressing the optimization challenges, it is important to structure the elements of problem variable in a specified manner that meet the needs of present challenges. The GA and PSO of this structure termed as chromosomes and particle position correspondingly. In BWO, it is termed as widow, each potential solution for a given problem is considered as black widow spider. In this case the optimization starts with spider initial population [15], then these are paired to reproduce another generation based on the cannibalism observations the female widow kills the male, she stores the male sperm and released in the eggs it gets hatch, give rise to young ones and they often disperse by the wind. This approach supports the optimization processes. Consider optimization problem of N^{var} dimension in terms of widow is an array of N^{var} be the solution of the optimization problem. The widow fitness F_t is get by computing the fitness function $(x^1, x^1, \cdots, x^{N^{\text{var}}})$

Let widow= $[x^1, x^1, \cdots, x^{N^{\text{var}}}]$, fitness= $F_t(x^1, x^1, \cdots, x^{N^{\text{var}}})$.

OBL based initialization process

The OBL is employing in many scenario of the present environment, which is based on opposition concept. Assuming a point $a(a_1, \cdots, a_z)$ in z dimensional space, $a^{n1} \in [p^{n1}, q^{n1}],\ n1 = 1, 2, 3\cdots, z$. The x opposition is evaluated in Eq. (9). It is suitable for choosing better unknown solution in random and opposite direction. This optimization improves the chance of selecting optimal region.

$$\breve{a}^{n1} = p^{n1} + q^{n1} - a_{n1} \tag{9}$$

Traditional Tent map calculation:

The tent map is a group of functions. Its repetitive function creates a discrete dynamic model. This tent map with some values of ck shown in Eq.(10) The tent map function with top at $x = \frac{1}{2}$ at a value of $\frac{1}{2}ck$. Each tent map region $0 \le x^n < \frac{1}{2}$ and $\frac{1}{2} \le x^n \le 1$ has one stable point.

$$x^{n+1} = T^{ck}(x^n) = \begin{cases} ckx^n & 0 \le x^n < \frac{1}{2} \\ ck(1 - x^n) & \frac{1}{2} \le x^n \le 1 \end{cases} \tag{10}$$

thereby

$$ck(1 - x^n) = x^n$$

$$1 - x^n = \frac{x^n}{ck}$$

$$1 = x^n(\frac{1}{ck} + 1)$$

$$\frac{ck}{ck + 1} = x^p$$

Similarly, the second region fixed point present only for $ck \ge 1$.

Initialization by TOBL strategy: Proposed Strategy

The proposed position initialization is done by combining the tent map and OBL [17], result in development of new strategy TOBL, which is expressed in Eq.(11) here, Z^t be the tent map. The value of Z^t be expressed in Eq.(12)

$$X^{ij} = lb^j + ub^j - Z^t * (ub^j - lb^j) \tag{11}$$

$$z^{t+1} = \left\{ \begin{array}{ll} \dfrac{z^t}{\beta_n} & 0.01 < z^t \le \beta_n \\ \dfrac{1-z^t}{1-\beta_n} & \beta_n < z^t \le 0.98 \end{array} \right\} \tag{12}$$

$$\beta_n = 0.5, z^t \in (0.02, 0.98)$$

The tent map and opposition based learning strategy are used for initialization phase of IBWO to increase the diversity of population.

4.2 Procreate: Proposed cross over operation

The pairs of widow spiders are reproduce lot of new ones, as per nature few ones have the chance to survive. In this algorithm, an array α_n has the ability to create long widow array consists of unsystematic numbers. The BWO approach has the ability to tackle incomplete convergent to a point. The array of IBWO is shown in Eq.(13) The crossover equation for IBWO is expressed in Eq.(14), here b^1 and b^2 are parents elements y^1 and y^2 are generated element, $iter$ be the repetitive counting of actual approach [18]. In the conventional approach of β_n array is fixed with 8 to improve the searching capacity of the algorithm.

$$\beta_n^{\,i} = 8 \times (\frac{2}{iter})^{0.3} \quad \beta_n = [\beta_n^{\,1}, \beta_n^{\,2}, \cdots, \beta_n^{\,n}] \tag{13}$$

$$\left\{ \begin{array}{l} y^1 = \alpha_n * b^1 + (1-\alpha_n) * b^2 \\ y^2 = \beta_n * b^2 + (1-\beta_n) * b^1 \end{array} \right\} \tag{14}$$

In our SI-BWO approach the value of β is updated and is expressed in Eq.(15),the value of β_n increasing gradually with increase of iteration and this will lead to the minimize the influence of local convergence.

$$\beta_n = e^{(\frac{t_i}{T_i})} - (\frac{t_i}{T_i}) \tag{15}$$

4.3 Cannibalism:

In this stage unfitted elements are rejected from the set and improves the speed of convergence. There are three methods of cannibalism in this optimization. Initial one is the female widow consumes the male during or after the mating. Likewise, based on the fitness value analyze the male and female in this optimization. Second is the sibling cannibalism, in this method stronger one defeat the weaker one. Based on CR, counting of fitted elements to be computed and the final method is to determine the optimal solution from the newly generated species..

4.4 Mutation: DE current to gbest mutation operator

In this algorithm, changes with key element is termed as mutation. The DE has three kinds of solution to solving a problem. They are target, donor, trial solutions. The target solution gives parent solution generating from the current iteration. The donor solution gives mutated solution produced through the differential operation and the target solution is a coupled solution of target and donor optimal solution.

Additionally, the DE-current to gbest mutation operator is employed in the mutation stage [19]. This mutation operator attains higher local improvement capacity and high speed convergence. These features affect the present state of the particle. The DE current to gbest mutation equation is shown in Eq.(16)where the value of α_n , k and F are shown in Eq.(17)

$$X^i(t) = \alpha_n . X^i(t) + F_s[X^*(t) + X^i(t)] + k[X^{r1}(t) - X^{r2}(t)] \qquad (16)$$

$$\alpha_n = 0.8 + 0.2 * (\frac{t_i}{T_i}) \; , \quad k = 0.1 + 0.2(\frac{t_i}{T_i}) \qquad (17)$$

In Eq.(16) F_s is stochastic scaling factor, which is inspired from levy flight. The levy flight is unsystematic order to estimate step size. Usually, $F_s \in (0,1]$. The scaling factor [16] F_s is computed using the updated equation which is expressed in Eq.(18)

$$F_s = step\,size \times ran(0,1) \qquad (18)$$

$$step\,size = \frac{s}{|w|^{1/\beta_n}} \; ,$$

here s and w are generated using normal distribution.

$$s \square N^1(0,\sigma_s^2) \; , \; w \square N^1(0,\sigma_w^2)$$

$$\sigma_s = \frac{\Gamma(1+\beta_n).\sin(\frac{\pi\beta_n}{2})}{\beta_n\Gamma[\frac{(1+\beta_n)}{2}].2^{(\beta_n-1)/2}}, \sigma_w = 1$$

The stochastic mutation scale factor is employed to improve exploitation ability of the DE mutation phase. . The flowchart of self improved black widow algorithm is shown in Figure 1.

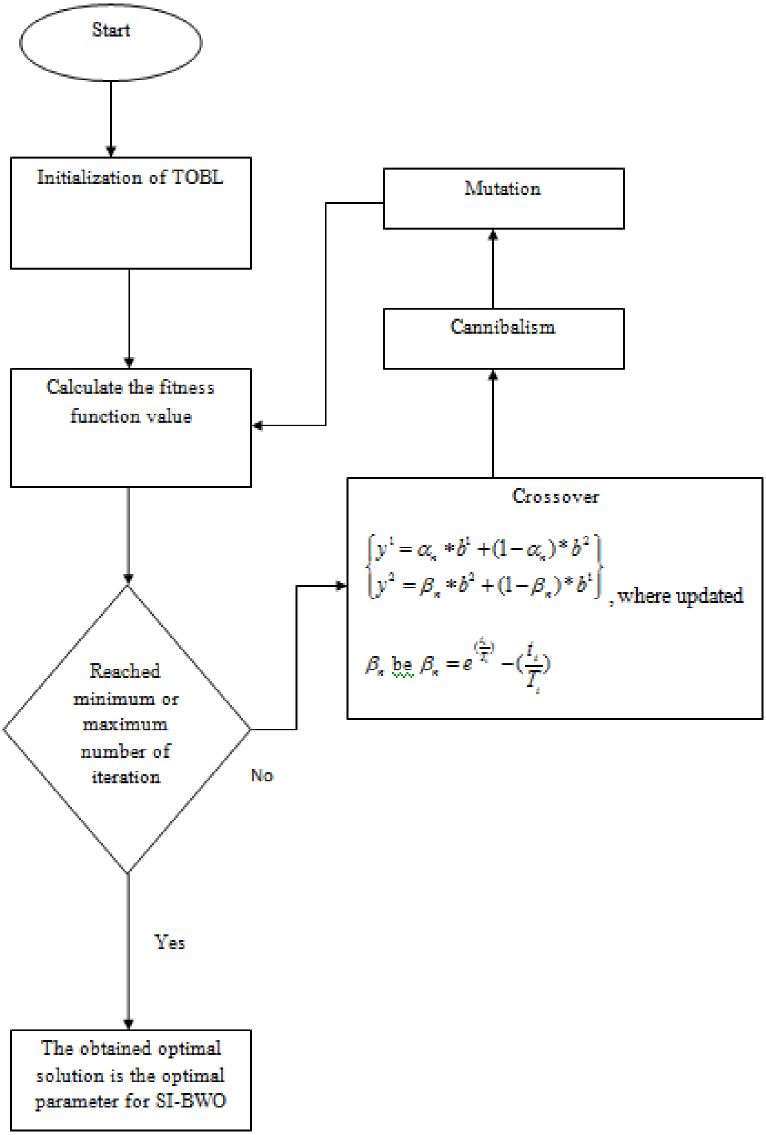

Start
Initialization of TOBL
Calculate the fitness function value
Mutation
Cannibalism
Crossover
$\begin{Bmatrix} y^1 = \alpha_\kappa * b^1 + (1-\alpha_\kappa) * b^2 \\ y^2 = \beta_\kappa * b^2 + (1-\beta_\kappa) * b^1 \end{Bmatrix}$, where updated
β_κ be $\beta_\kappa = e^{(\frac{t_i}{T_i})} - (\frac{t_i}{T_i})$
Reached minimum or maximum number of iteration
No
Yes
The obtained optimal solution is the optimal parameter for SI-BWO

Figure 1. Flowchart of self improved black widow optimization algorithm (SI-BWO)

5 Results and Discussions

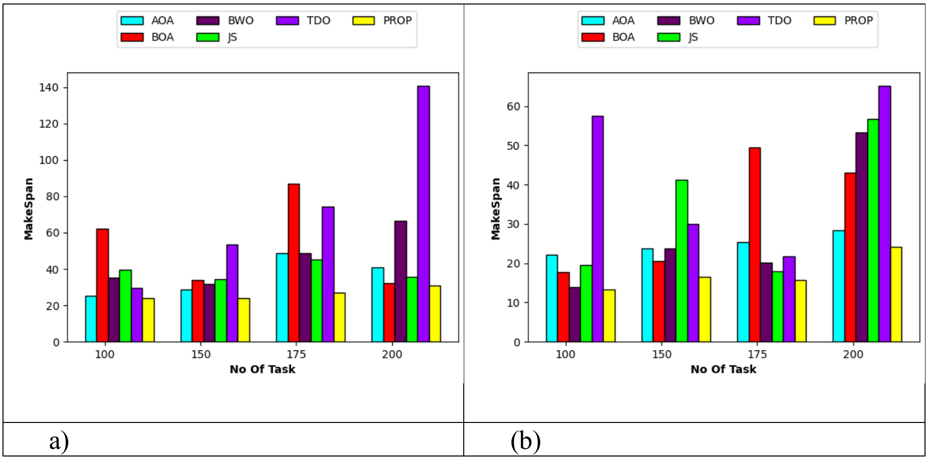
This work concentrates on cloud-based load balancing that is based on a multi-objective optimization model. Two environment specification is given: Cloud Environment-1 (one data centre with two real machines and ten VM beneath each PM). So in Cloud Environment-1 and Cloud Environment-2 (two data centres with four physical computers, and 10VM beneath each physical machine), there are a total of 20 VMs. In Cloud Environment-2, there are a total of 40 VM.

5.1 Performance Analysis

The performance of SIBWOA is evaluated in terms of makespan, memory utilization, migration cost, power consumption, response time, server load, turnaround time. Finally, the convergence analysis is also evaluated. Here, the conventional methods include (AOA)Arithmetic Optimization Algorithm, (BWO) Black Widow Optimization, (TDO) Tasmanian Devil Optimization, PROP in graph indicates the proposed algorithm, Self Improved Black widow optimization algorithm (SIBWO), Butterfly Optimization Algorithm (BOA) and (JS) Jelly Search

5.1.1 Makespan Analysis

Total time taken by the resources for the execution of all the tasks is considered to be the major factor, which must be low for the better performance and Figure 2 shows the performance graph. According to the analysis, two cloud environments are considered. In both the cases, the proposed SIBWOA shows the minimum makespan that is less than 30 seconds. even the number of tasks increases, the proposed model shows less makespan when compared to the conventional models. In cloud environment 1, even for 200 tasks, the makespan time is less for the proposed work. The Analysis on Makespan: proposed over conventional models is shown in figure 2.



a) (b)

Figure 2: Analysis on Makespan: proposed over conventional models

5.1.2 Analysis on Memory Utilization

The memory utilization for the requested service (task) must be low to determine that the given method is efficient. In this regard, Fig 4 shows the analysis of memory utilization for both proposed and conventional models. The analysis shows how the proposed algorithm requires least memory when compared to other conventional models to complete the tasks. The bars in the graph points that the proposed algorithm requires least memory even for 200 tasks that is less than 0.6, whereas the conventional methods need more memory to complete the same tasks. The same scenario is observed for the cloud environment 2. The Analysis on memory Utilization: proposed and conventional models is shown in figure 3.

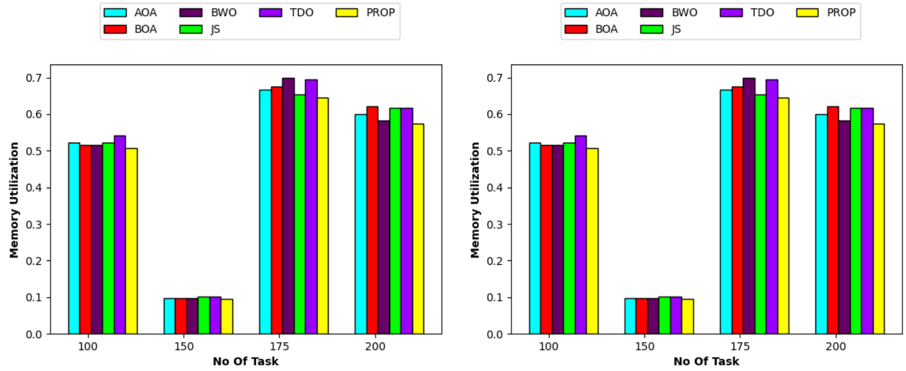


Figure 3: Analysis on memory Utilization: proposed and conventional models

5.1.3 Analysis on Migration cost

Migration, the process involves transfer of data from one storage to another that ensures that data are available in the right location at the right time to meet the need of the task. The cost required for the process must be less. Accordingly, the proposed algorithm ensures the least migration cost than the conventional models. While analysing the cloud environment 1, the proposed model needs only least migration cost of less than 0.0010 for 200 tasks, but the case of conventional models is different that shows poor performance. The same scenario is observed for cloud environment 2 (Figure 4).

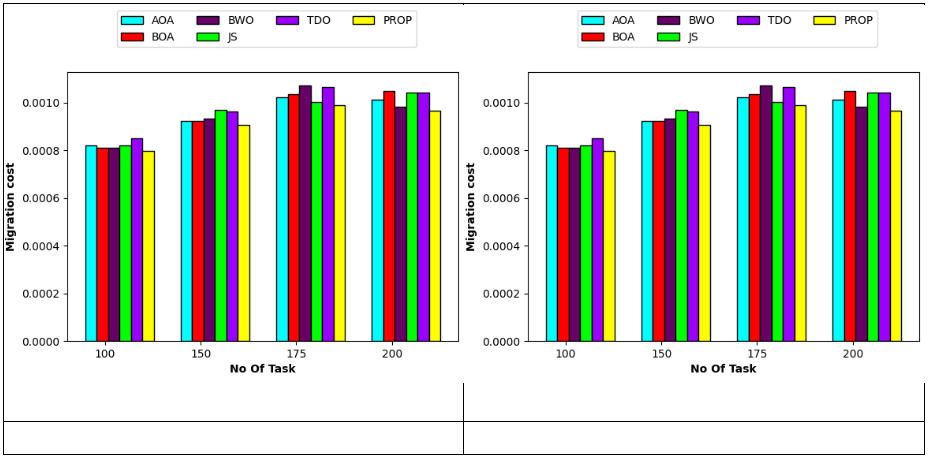


Figure 4: Analysis on Migration cost: Proposed and conventional models

5.1.4 Analysis on Power consumption

The load balancing with less power consumption is the greatest deal of the proposed optimization algorithm, whereas the other models are poor in this case that consumes more power to complete the tasks. The requirement of power increases as the tasks increases. However, for more tasks, than the conventional models, the proposed work consumes only less power to complete the tasks. This performance is obtained for both cloud environments (Figure5).

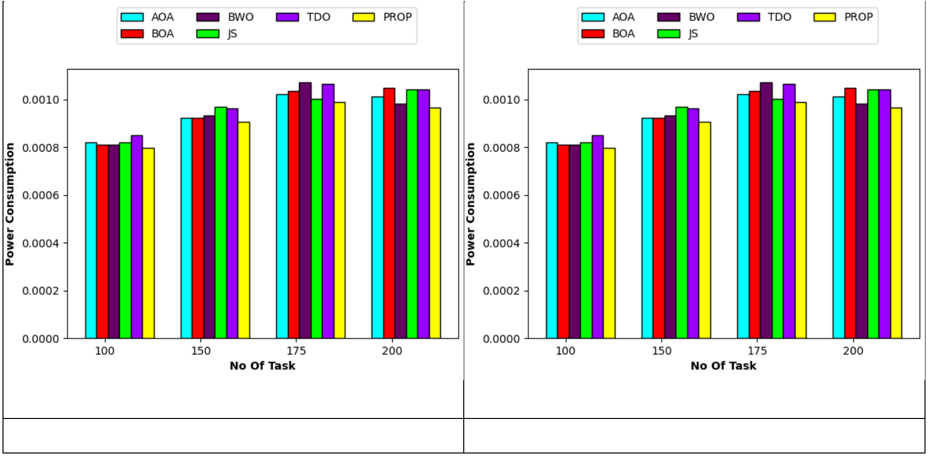


Figure 5: Analysis on power consumption: proposed and conventional models

5.1.5 Analysis on response time

Figure 6 explains the representation of performance of proposed work and other methods in terms of response time. The proposed model proves in this regard with less response time, however, the conventional methods incur more time for response almost in all tasks variation.

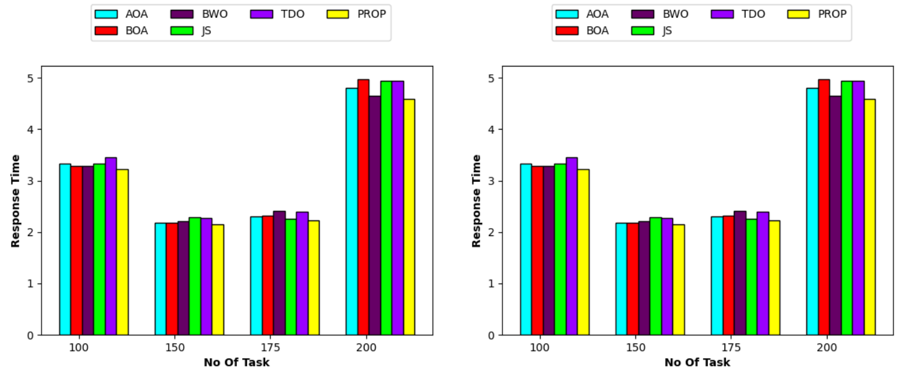


Figure 6: Analysis on response time: proposed and conventional models

5.1.6 Analysis on Server Load

The performance of proposed algorithm is proved in the server load case also over the conventional models as per the illustration in Figure 7. The variation in tasks show the performance efficiency of the proposed work for both cloud environment.

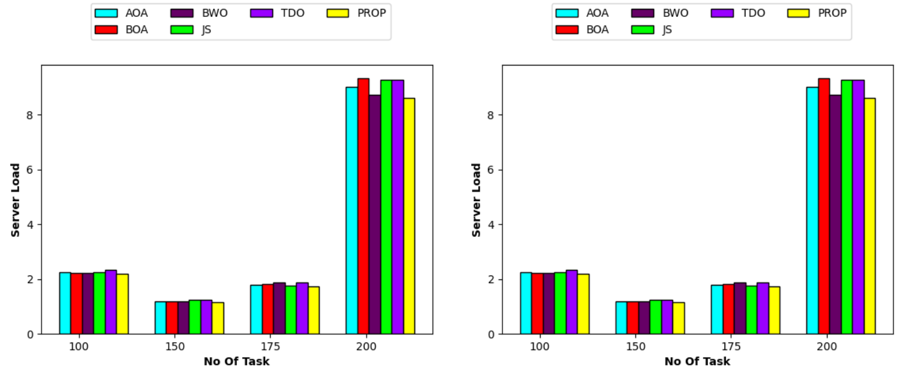


Figure 7: Analysis on Server load: proposed and conventional models

5.1.7 Analysis on turnaround Time

The turnaround time of the proposed work over the conventional models is evaluated and shown in Fig 8. The time when the process completes its execution must be minimal, and this has been attained by the proposed algorithm. The conventional models show their poor performance in this regard with more time taken to complete the execution. This has been attained by the strong architecture developed, SIBWOA.

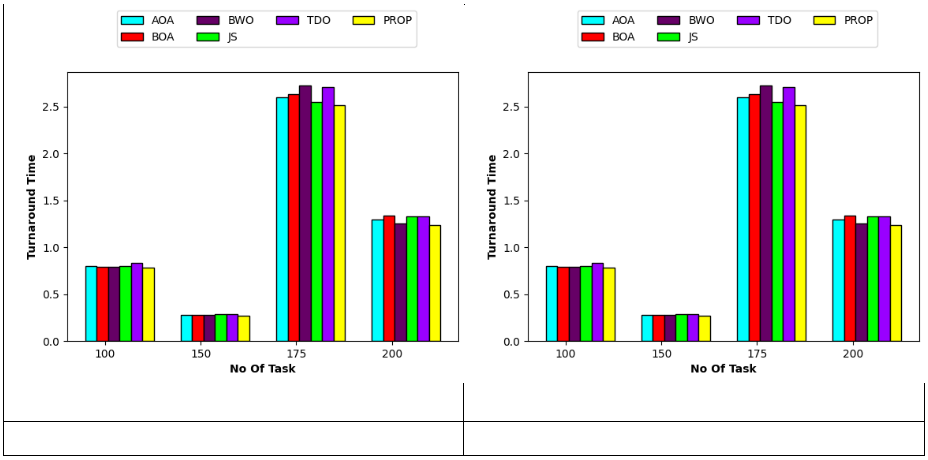


Figure 8: Analysis on Turnaround Time: proposed and conventional models

5.1.8 Convergence analysis

Figure9 shows the least convergence obtained by the proposed model than the conventional methods for both the cloud environment. The minimization of defined objective is attained by the proposed work when compared to the conventional models in both the cloud environments. Here, the iteration ranges from 0 to 100. The graphical representation reveals that the minimum convergence is attaining with the increase of iterations as when the iteration reaches to 100, the convergence rate reaches below 0.1, whereas the conventional methods shows high error rate. This has been observed in both the cloud environment.

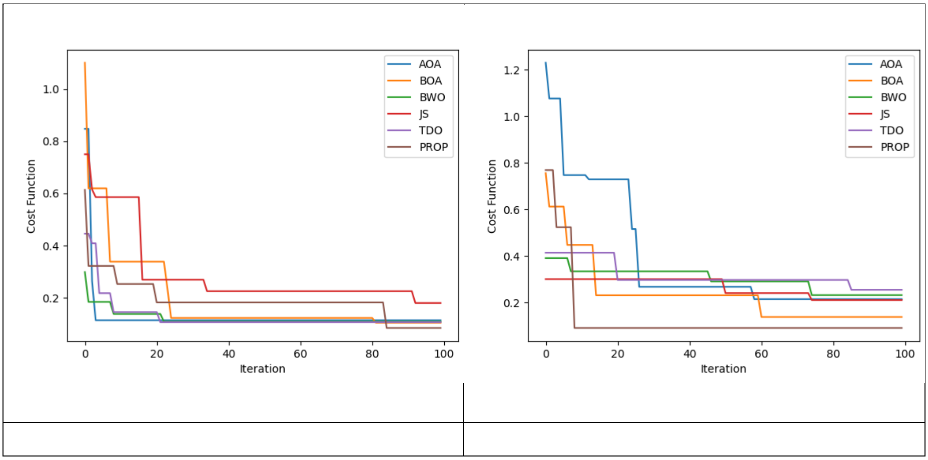


Figure 9: Convergence Analysis of proposed and conventional models

5.1.9 Statistical analysis

Table I shows the statistical evaluation of the performance of proposed model over the other techniques. The evaluation is pretending to be done under the estimation of cases like median, standard deviation, minimum value, maximum value. The evaluation is done in terms of error rate for both the cloud environment. The summarized value show the attainment of least error rate by the proposed work.

Table I: Statistical analysis of proposed and conventional models

	min	max	mean	median	std
Cloud Environment- 1					
AOA	0.113943	0.847413	0.130109	0.113943	0.103549
BOA	0.104375	1.100731	0.194437	0.122741	0.163865
BWO	0.108938	0.298715	0.120191	0.108938	0.027378
JS	0.180017	0.750005	0.290862	0.225329	0.1419
TDO	0.106661	0.445693	0.12893	0.106661	0.065691
PROP	0.084337	0.613675	0.189836	0.182233	0.073817
Cloud Environment-2					
AOA	0.214055	1.229217	0.380653	0.266992	0.258022
BOA	0.137362	0.754829	0.234943	0.230705	0.13116
BWO	0.231417	0.390217	0.29883	0.290286	0.047492
JS	0.210023	0.300033	0.262229	0.27003	0.039262
TDO	0.254213	0.41336	0.313497	0.29645	0.052064
PROP	0.090227	0.768262	0.132203	0.090227	0.146258

6 Conclusion

In this paper, a SI-BWO based load balancing model was introduced in cloud. When received the user request for task in the cloud, the PM assigned certain VM based on power consumption, migration cost, memory usage, and load balancing. Some parameters like response time, turn around time, and server load were considered for load balancing in cloud. To attain efficient load balancing, the optimal VM was selected by utilizing this proposed Si-BWO optimization for computing. Finallymakespan, memory usage, migration cost, power consumption, response timeserver load, throughput, turnaround time, and convergence of the proposed SI-BWO model were evaluated.

7 References

1. K. Dhana Sree Devi,D. Sumathi, Vignesh V, Chunduru Anilkumar, Kirankumar Katarakie,S. Balakrishnan,"CLOUD load balancing for storing the internet of things using deep load balancer with enhancedsecurity",Measurement: Sensors, vol 28,pp.100818,2023.

2. Imane Aly Saroit⇑, Dina Tarek,"LBCC-Hung: A load balancing protocol for cloud computing based on Hungarian method",Egyptian Informatics Journal, vol 24, pp.100387, 2023.
3. J.RobertAdaikalaraj, C. Chandrasekar,"To improve the performance on disk load balancing in a cloud environment using improved Lion optimization with min-max algorithm", Measurement: Sensors, vol 27, pp.100834, 2023.
4. Ajay Jangra, Neeraj Mangla,"An efficient load balancing framework for deploying resource scheduling in cloud based communication in healthcare",Measurement:Sensors, vol 25, pp.100584, 2023.
5. Mohammad Haris, Swaleha Zubair,"Mantaray modified multi-objective Harris hawk optimization algorithm expedites optimal load balancing in cloud computing",Journal of King Saud University – Computer and Information Sciences, vol 34, pp.9696–9709, 2022.
6. Hui-Ching Hsieh · Mao-Lun Chiang,"The Incremental Load Balance Cloud Algorithm by Using Dynamic Data Deployment",Journal of Grid Computing, vol 17,pp. 553–575,2019.
7. Chunlin Li, Jianhang Tang, Tao Mab, Xihao Yang, YoulongLuo,"Load balance based workflow job scheduling algorithm in distributed cloud",Journal of Network and Computer Applications, vol 152,pp.102518, 2020.
8. A. Francis Saviour Devaraj, Mohamed Elhoseny, S. Dhanasekaran, E. Laxmi Lydia, K. Shankar,"Hybridization of firefly and Improved Multi-Objective Particle Swarm Optimization algorithm for energy efficient load balancing in Cloud Computing environments",Journal of Parallel and Distributed Computing, vol 142,pp.36–45,2020.
9. Harvinder Singh, Sanjay Tyagi, Pardeep Kumar,"Cloud resource mapping through crow search inspired metaheuristic load balancing technique",Computers and Electrical Engineering, vol 93,pp.107221, 2021.
10. .Seyedhamid Mashhadi Moghaddam, Michael O'Sullivan, Charles Peter Unsworth, Sareh Fotuhi Piraghaj, Cameron Walker,"Metrics for improving the management of Cloud environments—Loadbalancing using measures of Quality of Service, Service LevelAgreement Violations and energy consumption", Future Generation Computer Systems, vol 123, pp.142–155, 2021.
11. Kumar, Voruganti Naresh, U. Sivaji, Gunipati Kanishka, B. Rupa Devi, A. Suresh, K. Reddy Madhavi, and Syed Thouheed Ahmed. "A Framework For Tweet Classification And Analysis On Social Media Platform Using Federated Learning." Malaysian Journal of Computer Science (2023): 90-98.
12. Arabinda Pradhan, Sukant KishoroBisoy,"A Novel Load Balancing Technique for Cloud Computing Platform based on PSO",Journal of King Saud University - Computer and Information Sciences, vol 34,pp.3988-3995,2022.
13. J. Gera, K. Sushma and S. R. Polamuri, "RECS Methodology for Secured Data Storage and Retrieval in Cloud," 2023 International Conference on Sustainable Computing and Data Communication Systems (ICSCDS), Erode, India, 2023, pp. 1426-1429, doi: 10.1109/ICSCDS56580.2023.10105033.
14. Altaf Hussain, Muhammad Aleem, Muhammad Azhar Iqbal,Muhammad Arshad Islam,"SLA-RALBA: cost-efficient and resource-aware load balancing algorithm for cloud computing",The Journal of Supercomputing, vol 75, pp.6777–6803, 2019.

15. VahidehHayyolalam, Ali Asghar PourhajiKazem,"Black Widow Optimization Algorithm: A novel meta-heuristic approach for solving engineering optimization problems",Engineering Applications of Artificial Intelligence, vol 87,pp.103249, 2020.
16. M. K. B, M. S. Kumar, F. D. Shadrach, S. R. Polamuri, P. R and V. N. Pudi, "A binary Bird Swarm Optimization technique for cloud computing task scheduling and load balancing," 2022 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2022, pp. 1-6, doi: 10.1109/ICSES55317.2022.9914085.
17. Weibin Kong, Yi Du, Xiaoyu Zhang, BaominJia,Zhongqing Fang, Qiong Liang, and Xue Du,"Multi-strategy enhanced coot algorithm for coverage optimization in wireless sensor networks",Electronics Letters, vol 59,2023.
18. Hui Liu, Guo Zhou, Yongquan Zhou,"Huajuan Huang1,2 and Xiuxi Wei1,2An RBF neural network based on improved black widow optimization algorithm for classification and regression problems",Front. Neuroinform, vol16, 2023.
19. Qiuyu Li and ZhitengMa,"A Hybrid Dynamic Probability Mutation Particle SwarmOptimization for Engineering Structure Design",Hindawi Mobile Information Systems, vol2,pp.32, 2021

Nomenclature:

Abbrevation	Descriptions
AO	Associated Overhead
BWO	Black widow optimization
CR	Cannibalism rating
DE	Differential Evaluation
DLB	Deep load balancer
FCFS	First come first serve method
IaaS	Infrastructure as aService
ILO	Improved Lion Optimization
LBCC-Hung	Load Balancing Protocol for Cloud ComputingBased on Hungarian Method
MRFO	Manta Ray Forging Optimization
MS	Makespan
MT	Migration Time
OBL	Opposition based learning
PaaS	Platform as a Service
PM	Physical Machine
RT	Response Time
SaaS	Software as a Service
SI-BWO	Self improved black widow optimization
VM	Virtual Machine