International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

Employing the Powered Hybridized Darts Game with BWO Optimization for Effective Job Scheduling and Distributing Load in the Cloud-Based Environment

[1]Manish Chhabra, [2]Dr.Rajesh E
[1]Research Scholar 21scse3010022
School of Computing Science & Engineering,
Galgotias University
Greater Noida, Uttar Pradesh 20320, India
email: manishchabra77@gmail.com
[2]Professor,
School of Computing Science & Engineering,
Galgotias University
Greater Noida, Uttar Pradesh 20320, India
email: rajsugan20@gmail.com

Abstract- One of the most frequent issues in cloud computing systems is job scheduling, which is designed to efficiently reduce installation time and cost while concurrently enhancing resource utilisation. Limitations such as accessible implementation costs, high resource utilisation, insufficient make-span, and fast scheduling response lead to the Nondeterministic Polynomial (NP)-hard optimisation problem. As the number of combinations along with processing power increases, job allocation becomes NP-hard. This study employs a hybrid heuristic optimisation technique that incorporates load balancing to achieve optimal job scheduling and boost service provider performance within the cloud architecture. As a result, there are many less problems with the scheduling process. The suggested work scheduling approach successfully resolves the load balancing issue. The suggested Hybridised Darts Game-Based Beluga Whale Optimisation Algorithm (HDG-BWOA) assists in assigning jobs to the machines according to workload. When assigning jobs to virtual machines, factors such as reduced energy usage, minimised mean reaction time, enhanced job assurance ratio, and higher Cloud Data Centre (CDC) resource consumption are taken into account. By ensuring flexibility among virtual computers, this job scheduling strategy keeps them from overloading or underloading. Additionally, by employing this method, more activities are effectively finished before the deadline. The effectiveness of the proposed configuration is guaranteed using traditional heuristic-based job scheduling techniques in compliance with multiple assessment metrics.
Keywords- Job Scheduling; Optimisation Problem; Cloud Computing; HDG-BWOA ; Cloud Data Center ;NP-hard;Traditional Heuristic

I. INTRODUCTION

Cloud computing is a relatively new concept that utilises existing internet technologies. Because traditional cloud computing uses a pay-per-use business model to provide platforms, applications, and infrastructure via the Internet, it is more similar to service-oriented computing [9]. “High-Performance Computing (HPC)” resources are combined with additional IT tools to create a single "Cloud Data Centre (CDC)" where HPC resources are ready to meet client demands [10]. In addition to requiring a lot of energy due to the intricacy of the hosts and equipment, this procedure has an effect on the environment[11]. To satisfy their needs for technology and data storage, a variety of companies, organizations, and educational institutions have chosen CDCs in recent years. [12]. The cloud computing paradigm allows the user to be freed from the information technology infrastructure [13]. The CDC's technical resources frequently include a large number of dependable, adaptable, and reasonably priced computers as well as cooling facilities. [14].

Massive data centers, vast amounts of storage, high-bandwidth networks, and other shared computing resources make cooperative computing possible. [15]. This procedure necessitates the efficient management of numerous machines in data centers, which makes cloud computing an improved computing paradigm. [16]. Among the phases of cloud computing are grid technology, service computing, parallel processing, and consolidated computing[17].In a recent research field, the job schedule in the data center is a critical process [18]. A well-functioning job scheduling system is essential for maintaining load balance in data centres. duty scheduling is regarded as a crucial duty since it can decrease running times and maximiser asset utilization. [19].It is impossible to fully achieve concurrency if jobs are not planned appropriately. Inconvenient scheduling results in lengthy implementation times, large costs, and insufficient resource use, all of which have an impact on the cloud structure's overall performance. [20]. Schedules that work are those that make better use of available resources. Scheduling techniques allocate work to virtual nodes in order to save costs and execution time. Cloud providers operate their services under the pay-per-use model. The "Quality of Service (QoS)" that users receive is determined by the "Service Level Agreement (SLA)." [21]. The data storage, connectivity, and response time of each activity are varied. A job breaks the SLA if it doesn't have the required resources. The cloud provider's QoS is managed if the SLA violation of the data center is high [22].

Researchers have recently become interested in the use of machine learning and meta-heuristic algorithms for job scheduling [23]. Meta-heuristic techniques like "Genetic Algorithm (GA)," "Particle Swarm Optimisation (PSO)," "Ant Colony Optimization (ACO)," and "Harmony Search (HS)" are frequently used to tackle multi-objective job scheduling problems. Because meta-heuristic algorithms take a long time to execute, there is an increase in processing overhead [24]. The problems with these methods are organized as follows: The execution time is lengthened at first, and the dynamic nature of cloud structures is beyond its capacity. Meta-heuristic techniques are required to verify the local and global optimal regions [25]. In the event that the algorithm unintentionally locates the local optimal zone, it swiftly advances to the global optimal region by trapping it. The traditional method of job scheduling incurs additional costs when executing the activities within a cloud virtual machine. Furthermore, reliability in the work arrangement process is not achieved by the conventional PSO-based job scheduling paradigm. In order to overcome these challenges, the researchers have proposed a novel hybrid meta-heuristic optimization approach. The signification contributions of the hybrid optimization-based job scheduling process are detailed as follows.

- The goal is to create a hybrid optimization-based job allocation model that assigns jobs to cloud virtual machines depending on capacity, allowing for quick job completion without compromising system performance.
- To offer the HDG-BWOA by integrating the DGO and BWO for optimizing the total amount of job that has to be arranged to the virtual machine of the cloud platform.
- To allocate the job to a proper virtual resource, an HDG-BWOA is recommended. It optimizes the job to be scheduled to the virtual resource for lowering the utilization of energy, mean reply time and resource employment and maximizing the job guarantee ratio during the job allocation process.
- By comparing the studied model's results with the methods already in use, the performance of the proposed hybrid optimization-based task scheduling model is examined in terms of different constraints..

The remaining parts of the HDG-BWOA-based job scheduling with optimal load balancing are described as follows. Part II lays up the framework of the current task allocation model, along with its applications and drawbacks. The theory underlying cloud computing's optimal load balancing work scheduling method is covered in section III. Part IV describes the creation of the hybridized darts game-based BWO as an effective task scheduling algorithm. Part V provides clarification on the goal function of work scheduling as well as a discussion of objective limitations. Part VI contains the results and discussion of the HDG-BWOA-based task scheduling with optimal load balancing. This paper is finally concluded in part VII. Literature survey

A. Related Works

A resource-efficient autonomous job scheduler based on supervised neural networks was proposed by Sharma and Garg [1] in 2020 with the aim of reducing makespan, energy consumption, operation overhead, and the total number of active units. The proposed "Artificial Neural Network (ANN)-based scheduler" used the current state of the cloud environment and an incoming job as inputs. Here, the genetic algorithm produced the enormous dataset. To assess the created model's effectiveness in cloud environments, it was compared to two other algorithms: "MinMIN-MINMin," a genetic algorithm, and "linear regression method-based energy efficient job schedulers." Results showed that the suggested work outperformed the current algorithms. A virtualized CDC's "Efficient Dynamic Scheduling Strategy (EDS)" was made available in 2021 by Marahatta et al. [2]. The initial identification of various jobs and virtual machines in the scheduling scheme was done in compliance with an earlier scheduling record. Next, similar-type jobs were grouped together to determine the host's functioning state. According to the findings, EDS greatly outperformed the earlier allocation techniques in terms of CDC resource usage, energy consumption mean reaction time, and job assurance ratio.. A job scheduling technique to lower energy expenses and the cloud's "Average Job Loss Probability (ATLP)" was proposed by Yuan et al. [3] in 2022. In this case, the energy cost of every cloud was minimized by combining the optimization of ATLP and the global mean. The management of the CDC and the distribution of employment to the web portal were regarded as the most crucial positions. However, in a cloud setting where variables like electricity costs and server accessibility displayed temporal variations, joint optimization was not achievable. Because of their coarse-grained nature, the QoS of jobs and the reduction of mean energy expenditure were not applied well in the current study. For real-time work scheduling, a novel technique known as adaptive "simulated annealing-based bi-objective differential evolution" was presented, which improved the power cost while preserving the quality of service. According to investigations, the suggested approach for task scheduling required less money to implement than the methods that were already in use. It could also be applied to automated manufacturing and smart cities etc.

A novel energy-efficient load-balancing global optimization technique was proposed by Lu and Sun [4] in 2019 to address the problem of energy usage in cloud computing. The resource-aware load clonal approach for job scheduling was another name for the created model. At first, the challenge of task scheduling was conceptualized as a multimodal optimization problem that sought to optimize energy usage as well as load distribution. In an era of sustainable cloud computing, the proposed algorithm may effectively reduce energy consumption, and its exploration and utilization capabilities may be improved and stabilized.

A hybrid optimization-based task scheduling method that efficiently schedules the jobs with the least amount of delay was presented by Sha and Santhosh [5] in 2020. In addition to these, the work scheduling procedure took into account the overall production time, execution time, waiting period, effectiveness, and utilization. The results of the simulation demonstrated that, in terms of several performance indicators, the suggested scheduling approach outperformed the conventional scheduling strategy.

In 2022, Kakkottakath *et al.* [6] have suggested a "Multi-Objective Hybrid PSO (MOHPSO)" to enhance the job scheduling process in internet-based computing. "Fuzzy Manhattan distance-based clustering" was employed to group the resources in the cloud. In order to enhance the investigation and searching abilities of the optimization algorithm and schedule the jobs in a cloud environment, the conventional "Search and Rescue Optimization Algorithm (SAR)" was hybridized with the prominent PSO. In addition, the scheduling procedure was carried out via empirical workflows with different jobs such as "Cybershake, Montage, and Epigenomics". The cloud scheduling workflow issue was simulated using the CloudSim tool. By contrasting the proposed methodology with several cutting-edge algorithms, the effectiveness of the developed model was confirmed.

In 2020, Sharma and Garg [7] have offered a "Harmony-Inspired Genetic Algorithm (HIGA)" to tackle the issue of energy-efficient job scheduling in CDC architecture. In order to efficiently detect both local and global optimal sectors without utilizing resources, the HIGA incorporated the exploration and exploitation characteristics of genetic algorithms. This led to rapid convergence. The suggested model was mainly designed to lower the workload and the energy consumption during the job scheduling process. The HIGA improved energy efficiency while using fewer resources. The rack components were turned off to lower the cooling energy. The stimulation outcome was evident that the proposed HIGA offered high energy savings and an increase in makespan with minimal computational overhead.

In 2021, Sohaib *et al.* [8] have recommended a hybrid-ant genetic algorithm for scheduling jobs. The suggested algorithm differentiated jobs and computer programs into smaller units and integrated the elements of the "genetic algorithm" and the "ant colony algorithm". Following job distribution, pheromones were integrated into virtual machines. The suggested technique effectively decreased the size of the solution space by partitioning the jobs into various groups and by determining the active virtual nodes. The faster convergence and response times were offered by the minimal solution space of the proposed algorithm. The developed model was used to lower the duration of workflows and jobs. The suggested technique reduced execution time and overall data center expenses.

B. Problem statement

The major problem in cloud data centers is considered energy-efficient job scheduling, where there is a need to minimize energy utilization and makespan for maximizing the performance of the cloud. This issue must be addressed, and thus, recent researchers are focused on designing new energy-efficient job scheduling approaches with the adoption of intelligent algorithms like meta-heuristic and deep learning approaches. However, these techniques also suffer from higher execution time or overhead due to the redundant solutions. The recent job allocating approaches in the cloud are reviewed in Table I. The genetic algorithm [1] performance of the designed model is enhanced by achieving lower execution overhead energy consumption, enhancing the makespan and reaching higher training accuracy by maintaining a lesser number of active racks. It is not suitable for allocating workflow applications with a group of individual jobs. Dynamic scheduling scheme [2] minimizes energy consumption by efficient dynamic job scheduling and also reduces the mean response time, improves the job guarantee ratio, and improves the overall scheduling efficiency. Although it achieves superior performance in job scheduling, it does not perform job failure predictions. Adaptive simulated-annealing-based bi-objective differential evolution [3] achieves lower energy cost and speed of convergence, enhances the diversity of Pareto-optimal individuals and reduces dynamic energy consumption. It suffers from processing complicated scenarios with more number of cloud data centers. The CSRSA [4] minimizes maintenance and operation costs, minimizes heat generation, and also reduces energy consumption, and, efficiently performs load balancing and saves energy. It is not applicable for large scale data centers. Hybrid optimization [5] has achieved superior efficiency regarding utilization, efficiency, waiting time, execution time and overall production time and the overall performance of the job scheduling approach is enhanced. It suffers from convergence issues. MOHPSO [6] achieves higher performance improvement regarding metrics like cost, makespan, and load balancing. It faces complications regarding a lower convergence rate. HIGA [7] reduces the execution overhead, improves the application performance regarding makespan, and reduces the execution time complexity. It does not solve the temperature effects. Hybrid and genetic algorithm [8] reduces the running time of the jobs and workflows and also minimizes the response time and convergence time. The dependencies among jobs are not addressed. This review helps us to suggest a new job scheduling approach in a cloud with the aim of suggesting novel techniques.

II. THE PRINCIPLE BEHIND THE JOB SCHEDULING PROCESS WITH OPTIMAL LOAD BALANCING IN CLOUD COMPUTING

A. Job Scheduling in Cloud Computing: Problem Formulation

The cloud system consists of several nodes, and they are associated with high heterogeneity and complexities. The total amount of jobs is raised with an increase in the count of users in the cloud platform. In the cloud era, efficient job allocation is a complex issue. The safety administration, resource organization, user administration and job supervision is the important consideration of the cloud platform layer. The cloud software application and communal interfaces are integrated into the cloud application layer. The job allocation process splits the job if the volume of the job is too high. Initially, the sub-jobs are generated by splitting the job given by the user. Further, the job and the virtual machine are mapped together by considering the difference of sub-jobs. Some specific techniques is considered to arrange the job o to the heterogeneous virtual nodes represented as n . The sub-job derived from the main jobs given by the user is indicated as $U = \{U_1, U_2, .. U_o\}$. These sub-jobs are assigned to the virtual machines of the cloud that is represented as $W = \{W_1, W_2, \ldots W_n\}$. The virtual machine and the job are mapped for restricting the execution of sub-jobs in various virtual machines, and it is indicated in Eq. (1).

$$UW_{mapping} = \begin{Bmatrix} U_1W_1 & U_2W_1 & \cdots & U_oW_1 \\ U_1W_2 & U_2W_2 & \cdots & U_oW_2 \\ \vdots & \vdots & \vdots & \vdots \\ U_1W_n & U_2W_n & \cdots & U_oW_n \end{Bmatrix} \tag{1}$$

Here, the mapping in the middle of the virtual nodes and the job is represented as U_oW_n . The QoS, proper resource consumption and tiny implementation time are achieved by this job allocation process.

The fundamental structure of job allocation in a cloud platform is depicted in Fig. 1.

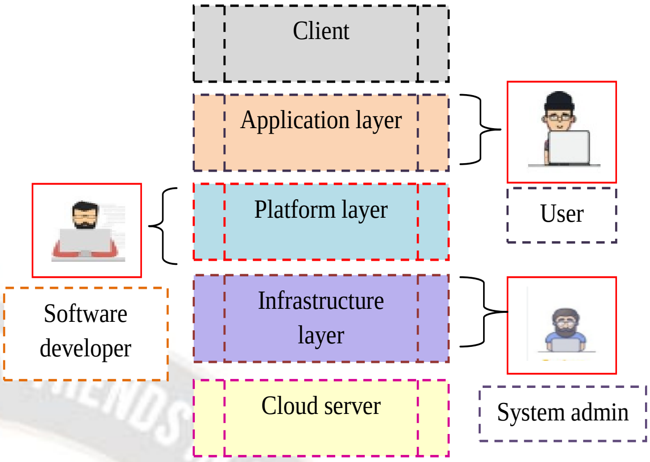


Figure 1. Basic structure of resource allocation in cloud platform

TABLE I. USES AND LIMITATIONS OF TRADITIONAL JOB SCHEDULING APPROACHES IN THE CLOUD

Author [citation]	**Methodology**	**Features**	**Challenges**
Sharma and Garg [1]	Genetic algorithm	• The performance of the designed model is enhanced by achieving lower execution overhead energy consumption and enhancing the makespan. • It reaches higher training accuracy by maintaining a lesser number of active racks.	• It is not suitable for allocating workflow applications with a group of individual jobs.
Marahatta *et al.* [2]	Dynamic scheduling scheme	• This model minimizes energy consumption by efficient dynamic job scheduling and also reduces the mean reaction instance, improves the job assurance ratio, and improves the overall scheduling efficiency.	• Although it achieves superior performance in job scheduling, it does not perform job failure predictions.
Yuan *et al.* [3]	Adaptive simulated-annealing-based bi-objective differential evolution	• It achieves lower energy cost speed of convergence and enhances the diversity of Pareto-optimal individuals. • It reduces the dynamic energy consumption.	• It suffers from processing complicated scenarios with more number of cloud data centers.
Lu and Sun [4]	Clonal Selection Resource Scheduling Algorithm (CSRSA)	• It minimizes the maintenance and operation costs, minimizes the heat generation, and also reduces energy consumption. • It efficiently performs the load balancing and saves energy.	• It is not applicable for large scale data centers.
Sha and Santhosh [5]	Hybrid optimization	• It has achieved superior efficiency regarding utilization, efficiency, waiting time, execution time and overall production time. • The overall performance of the job scheduling approach is enhanced.	• It suffers from convergence issues.
Kakkottakath*et al.* [6]	MOHPSO	• It achieves higher performance improvement regarding metrics like cost, makespan, and load balancing.	• It faces complications regarding a lower convergence rate.
Sharma and Garg [7]	HIGA	• It reduces the execution overhead and improves the application performance regarding makespan. • It reduces the execution time complexity.	• It does not solve the temperature effects.
Sohaib*et al.* [8]	Hybrid ant genetic algorithm	• It reduces the management time of the jobs and workflows. • It also minimizes the response time and convergence time.	• The dependencies among jobs are not addressed.

B. Proposed Job Scheduling Model in Cloud Computing

The efficiency of the cloud structure is enhanced by the job allocation process. In the cloud environment, profit can be gained by dispersing the jobs to the virtual resources. The conventional techniques utilize a huge quantity of resources for the job allocation process. Furthermore, the conventional technique takes more cost and time for the job allocation process, so it isn't easy to fulfill the requirements of the clients. Besides, the handling time of the conventional job scheduling model is too high. In addition, the previous technique does not consider important attributes like reliability and availability in the job arrangement process. Thus, it affects the effectiveness of the cloud-based services. Existing techniques overutilize the resources, which leads to the degradation of cloud service performance. The conventional job scheduling models take more time to distribute the complex jobs to the virtual resource of the cloud. Moreover, existing techniques do not have the capacity to identify the sequence of jobs, so it does not satisfy the requirements of the customer in the stipulations of QoS. Thus, the intelligent heuristic-based job scheduling model in the cloud era is designed to fix the difficulties. The structural outline of the heuristic-assisted job scheduling framework is characterized in Fig. 2.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

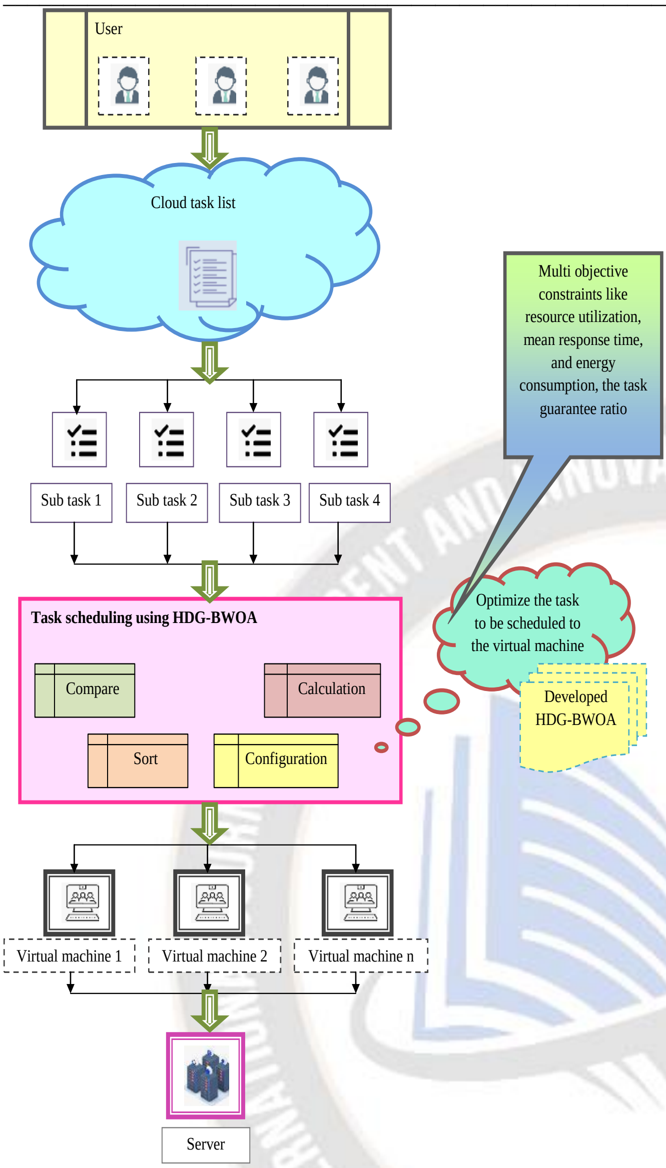


Figure 2. Structural outline of the heuristic-assisted job scheduling framework

The hybrid heuristic-assisted job scheduling model is offered to assign the jobs to the appropriate virtual nodes of the cloud so that the job can be completed within a particular time without affecting the efficacy of the system. The developed job allocation approach assigns the top-priority job to the cloud for rapid execution. The suggested job scheduling model also appraises the load of the virtual nodes in the cloud for scheduling the jobs based on the capacity of the virtual machine. The user can give an assortment of jobs to the cloud structure. The received jobs are available in various sizes and types. After that, the collected job from the user is divided into sub-jobs based on the importance and execution time. After dividing the collected job into subjobs, the offered HDG-BWOA is grasped to allocate the job to the virtual resource of the cloud. In this process, the recommended HDG-BWOA optimizes the job to be allocated for the virtual machine; hence, the "mean response time, resource utilization, and energy consumption" is lowered. Additionally, the "job guarantee ratio" is highly maximized by the optimization process. The developed job scheduling algorithm considers the adequacy of the virtual resource, so the overloading action of the virtual resource is greatly neglected. Moreover, the job can be finished within a specified period because of the suggested job scheduling model. At last, the results of the p[resented model are correlated with the existing approaches to find the usefulness of the explored job scheduling with the load balancing model.

III. DEVELOPMENT OF HYBRIDIZED DARTS GAME-BASED BELUGA WHALE OPTIMIZATION AS EFFICIENT JOB SCHEDULING ALGORITHM

A. DGO

The DGO [26] is the game-based optimization algorithm that mainly mimics the rules of the darts game.

The darts game is a simple game, and it is played by all genders and all age people. The dashboard and darts are the most important tools for the darts game.

Arithmetic design: The matrix is employed to design the population of players. In this matrix, the traits of the player is indicated in the column of the matrix, and the players are indicated in the row of the matrix. Here, the count of the problem variable and the count of columns in the matrix is considered as same.

The useful information is obtained by placing the term Y_j in the fitness function, and it is specified in the below expression.

$$G_{BEST} = mimum(FIT)_{O\times 1} \tag{2}$$

$$W_{BEST} = W(place\ of\ \ mimum(FIT), 1:n) \tag{3}$$

$$G_{WORST} = mximum(FIT)_{O\times 1} \tag{4}$$

$$W_{WORST} = W(place\ of\ \ mimum(FIT), 1:n) \tag{5}$$

$$G^{o} = \frac{FIT - G_{WORST}}{\Sigma_{k=1}^{O}(FIT_k - G_{WORST})} \tag{6}$$

$$Q_j = \frac{G_j^o}{mxi(G^o)} \tag{7}$$

Thus, the excellent variable is pinpointed as W_{BEST}, the best fitness function is elucidated as G_{BEST}, for the player j, the probability function is termed as Q_j, the normalized fitness function is signified as G^o, the worst fitness function is pinpointed as G_{WORST}, and the worst variable is represented as W_{WORST}.

The dartboard consists of diverse scores with a total of 82 areas. At each iteration process, the player can throw the darts in three times. The player's skill and the chance are used to analyze the location of the darts in the dartboard.

For each player, the throwing score is identified by Eq. (8) to Eq. (9).

$$D_j = circle(82 \times (1 - Q_j)) \quad (8)$$

$$TD_j = \begin{cases} T(1:D), \; rdnu < Q_j \\ T(D+1:82), \; else \end{cases} \quad (9)$$

$$t_j = TD_j(l) \text{ and } 1 \le l \le 82 \quad (10)$$

$$t_j^o = \frac{\sum_{flip}^{3} t_j^{flip}}{180} \quad (11)$$

Here, for the player j, the score for each hurl is represented as t_j, score candidates are indicated as TD_j, the normalized score of the player is delineated as t_j^o, the score matrix T is arranged into lower order.

The values of the issue variable and the innovated status of the player are indicated in Eq. (12).

$$W_j = W_j + rdnu(1,n) \times \left(W_{BEST} - 3t_j^o W_j\right) \quad (12)$$

Here, the innovated status of the player is represented as W_j.

The pseudocode of the DGO is elucidated in Algorithm 1.

Algorithm 1: DGO
Begin
Develop the primary population of the player
Find the fitness value
Upgrade the W_{BEST}, W_{WORST}, G_{WORST}, G_{WORST} by Eq. (2) to Eq. (5).
Upgrade the Q_j, and G^o using Eq. (6) and Eq. (7).
Find the t_j^o using Eq. (8) to Eq. (11).
Upgrade the W_j using Eq. (12).
Get best solution
End

B. BWO

The BWO [27] is the swarm-based Meta optimization algorithm. It is modeled on the basis of the behavior of the beluga whales. The optimization problem is easily solved by the BWO. The BWO is implemented in three phases, namely investigation, exploitation and whale fall.

Motivation: The beluga whale mostly lives in the sea, and it produces diverse sounds, so it is referred to as the canary of the sea. The adult whales are available in white color. The weight of the beluga is about 1500kg, and the length of the beluga whale is around 3.5 to 5.5m. The hearing and vision capabilities of the beluga are too sharp. The beluga whales produce diverse sounds for the hunting process. The beluga whales are mostly distributed in the Arctic and subarctic regions of the ocean. In some places, the beluga whales are sheltered in aquariums.

Arithmetical design of the BWO: The swimming, hunting and the whale fall activities of the beluga whale is employed for modeling the BWO. The investigation and the exploitation phases are also available in the BWO. The global searching capability of the beluga whale is modeled in the investigation of the BWO. Consequently, the local searching capability of the beluga whale is elaborated in the exploitation phase of the BWO. In this BWO, the beluga are examined as the search agents, and they adopt their corresponding position vector for causing displacement in the search area. The spot of the beluga whales is altered by the prospect of the whale falling in the BWO. In the BWO, the search agents are represented in the matrix W and the fitness value is indicated in the matrix E_Y.

The balance factor A_g is used by the BWO for converting the investigation to the exploitation stage, and it is illustrated in Eq. (13).

$$A_g = A_0\left(1 - \frac{S}{2S_{\max}}\right) \quad (13)$$

Thus, the value of the balance element A_g is greater than 0.5, it loads to the development of the exploration stage in BWO. The excellent iteration is constituted as $S_{\max}$, the present iteration is signified as S, at every iteration process, the value of A_0 is altered in the middle of $[0,1]$. If the value of $A_g \le 0.5$, then the exploitation stage is implemented. The value of A_g decreases from $[0,1]$ to $[0,0.5]$, because of an increase in iteration S. When the value of iteration S is augmented, it loads to an increase in the probability of the exploitation phase.

Investigation phase: The swimming traits of the beluga are used to examine the exploration phase of the BWO. Under the diverse position, the beluga whale performs the social sexual behavior. The mirrored manner of the two beluga is employed to determine the location of the search negotiator. The location of the beluga is upgraded based on the above consequences, and it is signified in Eq. (14).

$$\begin{cases} W_{j,k}^{s+1} = W_{j,O_k}^{s} + \left(W_{q,O_1}^{s} - W_{j,O_k}^{s}\right)(1+q_1)\sin(2\pi q_2), \; k = even \\ W_{j,k}^{s+1} = W_{j,O_k}^{s} + \left(W_{q,O_1}^{s} - W_{j,O_k}^{s}\right)(1+q_1)\cos(2\pi q_2), \; k = odd \end{cases} \quad (14)$$

Thus, for the beluga whale j and at the dimension k, the fresh spot of the beluga whale is represented as $W_{j,k}^{s+1}$, at the dimension c, the selected arbitrary number is represented as O_k, and $k = 1,2..c$. At the dimension O_k, the location of the beluga j is expressed as W_{j,O_k}^{s}. The existing position of j^{th} and q^{th} beluga whale is represented as W_{j,O_k}^{s}, and W_{q,O_1}^{s} correspondingly. The random number in the middle $[0,1]$ is represented as q_1, and q_2 respectively. The average fins of the reflected beluga in the direction of the facade is illustrated as

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

$\sin(2\pi q_2)$, and $\cos(2\pi q_2)$ respectively. The mirroring trait of the beluga whales while in the swimming position is reflected by the position value updated by the odd and even numbers. The erratic operator in the exploitation phase is enriched by the two random numbers q_1 and q_2, respectively.

Exploitation phase: The poaching characteristics of the beluga whale are designed in the exploitation stage of the BWO. The poaching action of the beluga whales is done in a cooperative manner, which means the information is passed to the adjacent best whale for executing the poaching process. The convergence rate in the exploitation stage of the BWO is enhanced by introducing the levy flight operator. This operator is used by the beluga whales for catching the target prey, and it is elucidated in Eq. (15).

$$W_j^{S+1} = q_3 W_{BEST}^S - q_4 W_j^S + B_1 \cdot K_G \left(W_q^S - W_j^S\right) \tag{15}$$

Here, the new spot of the beluga j is represented as W_j^{S+1}, the present iteration is represented as S, and the existing location of the arbitrary beluga is characterized as W_q^S. The existing spot of the beluga j is indicated as W_j^S, the arbitrary number is depicted as q_3, and q_4 respectively, and it is lies in the middle of $[0,1]$, the best position of the beluga whale is indicated as W_{BEST}^S, and the intensity of the levy flight is identified by the arbitrary leap durability and it is indicated in Eq. (16).

$$B_1 = 2q_4\left(1 - \frac{S}{S_{\max}}\right) \tag{16}$$

Eq. (17) gives the expression for the levy flight operator K_G.

$$K_G = 0.05 \times \frac{t \times \varpi}{|u|^{\frac{1}{\rho}}} \tag{17}$$

$$\varpi = \left(\frac{\Im(1+\rho) \times \sin\left(\frac{\pi\rho}{2}\right)}{\Im\left(\frac{(1+\rho)}{2}\right) \times \rho \times 2^{\left(\frac{\rho-1}{2}\right)}}\right)^{\frac{1}{\rho}} \tag{18}$$

Here, the absent factor is represented as ρ, and it is taken as 1.5; the usually dispersed arbitrary numbers are indicated as t, and u correspondingly.

Whale fall: The beluga whale is attacked by the polar bear, humans and other killer whales during the poaching and migration process. The beluga whale passes the information to other whales and intelligently escapes from the threats. Sometimes, the beluga whales fall on the sea bed because of the threats caused by the killer whales. The falling of whales under the sea bed is known as whale fall. The fallen body of the whale is consumed by the sharks and other animals in the sea. The remaining bones of the beluga whale are consumed by the bacteria and coral reefs of the ocean. The location of the beluga whales is updated by considering the dimension of the step in the whale fall action, the dimension of the population constant, location of the beluga are represented in Eq. (19).

$$W_j^{S+1} = q_5 W_j^S - q_6 W_j^S + q_7 W_{step} \tag{19}$$

Here, the dimension of the step in the whale fall process is represented as W_{step} and the random number is represented as q_5, q_6 and q_7 in $[0,1]$. The value of W_{step} is determined by Eq. (20).

$$W_{step} = (t_c - k_c)\exp\left(-B_2 \frac{S}{S_{\max}}\right) \tag{20}$$

Here, the upper and the lower boundary of the variable are represented as t_c *and* k_c, respectively, the chance of whale fall and the size of the population is mainly rely on the step factor B_2, and it is identified by Eq. (21).

$$B_2 = 2V_g * m \tag{21}$$

Here, the chance of whale fall is represented as V_g, and it is identified by Eq. (22).

$$V_g = 0.1 - 0.05\frac{S}{S_{\max}} \tag{22}$$

Here, the chance of whale fall is calculated in linear form, and its values is taken as 0.1 at the primary iteration. Further, the chance of whale fall is turned down to 0.05 at the final iteration. The pseudocode of the conventional BWO is given in Algorithm 2.

Algorithm 2: BWO
Initialize the population and maximum iteration size
While $S \le S_{\max}$ do
Find the balance factor A_g using Eq. (13).
Get the chances of whale fall using Eq. (22).
For entity beluga W_j do
If $A_g(j) > 0.5$
Execute the investigation phase and update the position of the beluga whale using Eq. (14).
Else if $A_g(j) > 0.5$
Execute the development phase and upgrade the location of the beluga whale using Eq. (15).
End if
Verify the boundary condition of the new position and find the fitness values
End for
For individual beluga
Execute the whale fall action
If $A_g(j) \le W_g$
Find the step factor B_2 using Eq. (21).

			Evaluate the step size W_{step} using Eq. (20).
			Verify the boundary condition of the new spot and find the fitness values
		End if	
	End for		
	Calculate the present best solution.		
	$S = S + 1$		
	End while		
Get best solution			

C. Developed HDG-BWO

The developed HDG-BWO is formed by incorporating the conventional DG and BWO. The total amount of job that are allocated to the virtual resource is optimized by the HDG-BWO. So it loads to the minimization of "resource utilization", "mean response time", and "energy consumption" in the job-scheduling process. In addition to that, the job guarantee ratio is maximized as a consequence of the optimization process. The overall processing time for the virtual machine is greatly turned down by the job allocation process. The existing BWO is the swarm-aided algorithm that effectively conquers the optimization problem, and the convergence rate and robustness of the BOW are very high. Yet, the discrete issues are not solved by the BWO, and it does not give effective optimization solutions for the big data application field. The DGO is a game-based optimization, and it is simple to understand. In addition, the exploration and exploitation capabilities are really well. Yet, the global optimal solutions are not solved by the DGO. So, the new HDG-BWO is developed to crack the aforementioned difficulties. The new HDG-BWO is executed by updating the location using the upcoming conditions. The maximum iteration and the current iteration are indicated as $S_{\max}$, and S respectively. If the present iteration S is divisible by 7 and 9, then the position W_j is upgraded by the DGO. Otherwise, the position W_j^{S+1} is updated by the BWO. The developed HDG-BWO can easily provide a globally optimal solution due to the position updating process. In addition, the discrete issues are efficiently handled by the developed HDG-BWO.

The pseudocode of the developed HDG-BWO is given in Algorithm 3, and the flow indication is shown in Fig. 3.

Algorithm 3: HDG-BWO		
Initialize the quantity of population $popu$ and greatest number of iteration $S_{\max}$		
For $u = t\ to\ S_{\max}$		
	For $j = 1\ to\ popu$	
		Upgrade the W_{BEST}, W_{WORST}, G_{WORST} and G_{WORST} by Eq. (2) to Eq. (5).
		Find the balance factor A_g using Eq. (13).
		Get the chances of whale fall using Eq. (22).

		If S is divisible by 7 and 9	
			Modernize the position using W_j Eq. (12) of DGO.
		Else	
			Upgrade the potion using W_j^{S+1} using Eq. (15) of BWO.
		End if	
	End for		
End for			
Get best solution			
End			

IV. Objective Function for Job Scheduling using Hybridized Darts Game-based Beluga Whale Optimization Algorithm

A. Objective Function of Job Scheduling

The objective function of job allocation is to lower the resource consumption, mean response time, and energy consumption in the job scheduling process and it is offered in Eq. (23).

$$obf = \underset{\{T_n^{VM}\}}{\arg\min}\left(RU + RT + EC + \frac{1}{TGR} \right) \tag{23}$$

Here, the job that has to be arranged to the virtual resource of the cloud is represented as T_n^{VM}. The resource utilization is elucidated as RU, the mean reaction time is pinpointed as RT, energy consumption is represented as EC and the job guarantee ratio is indicated as TGR.

The execution time must be minimal to execute the load balancing process, and the formula for the degree of imbalance is given in Eq. (24).

$$EJ = \frac{U(\max) - U(\min)}{U(avg)} \tag{24}$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

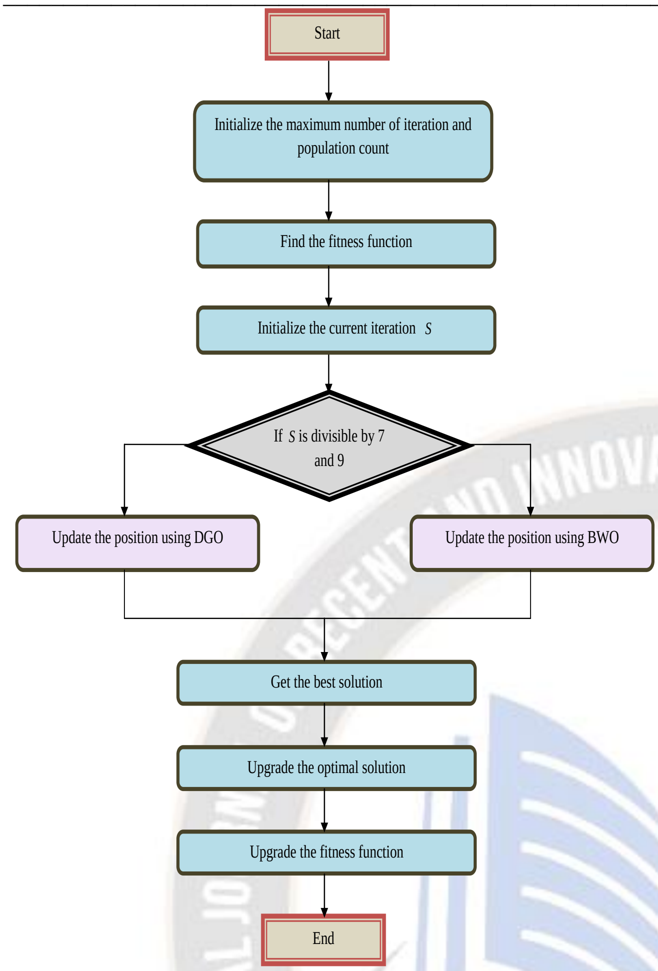


Figure 3. Flowchart of developed HDG-BWO

Here, the maximum, average and minimum execution times are represented as $U(\max)$, $U(avg)$, and $U(\min)$.

The solution illustration for the job scheduling process is denoted in Fig. 4.

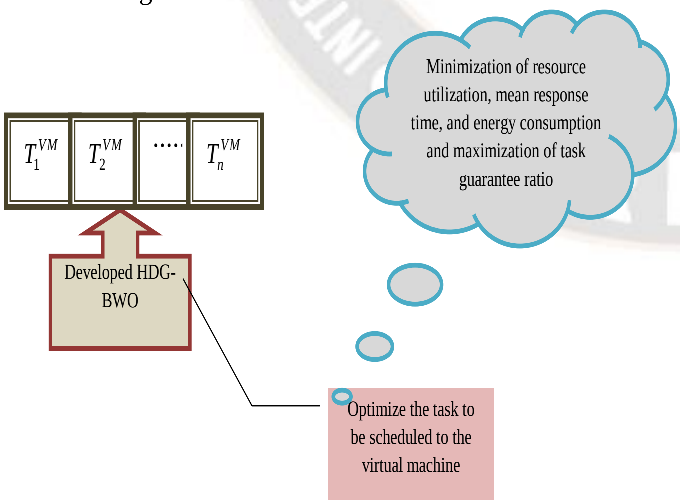


Figure 4. Solution diagram of the job scheduling process

B. Description of Job Scheduling Objective Constraints

The objective constraints in the job allocation process are elaborated as follows.

Resource Utilization: It states the total amount of resources utilized in the job allocation process, and it is offered in Eq. (25).

$$RU = \frac{\sum_{j=1}^{n} DU_j}{Makespan * n} \tag{25}$$

Here, the job completion time is represented as DU and the total count of virtual machines is specified as n.

Mean response time RT: The time consumed by the load balancing model in the cloud architecture is elucidated as mean response time, and it is given in Eq. (26).

$$RT = \sum_{j=1}^{o} DU + TC \tag{26}$$

Here, the term TC denotes the capitulation time of the job and job completion time is represented as DU.

Energy consumption: It is defined as the summation of energy used in the job allocation process (active energy) and the energy utilized by the virtual machine when it is in an inactive state (inactive energy). This is expressed in Eq. (27).

$$EC = EC(act) + EC(inact) \tag{27}$$

Here, the active energy is represented as $EC(act)$ and the inactive energy is represented as $EC(inact)$.

Job guarantee ratio: It is the ratio of the throughput to the total number of completed jobs, and it is signified in Eq. (28).

$$TGR = \frac{throughput}{TNCT} \tag{28}$$

Here, the term $TNCT$ represents the total number of completed jobs.

V. RESULTS AND DISCUSSION

A. Simulation setup

The HDG-BWO-based job allocation model with optimal load balancing was tested in the Python platform. In this analysis process, the maximum iteration and number of population were taken as 250 and 10, correspondingly. Here, the chromosome length was taken the same as a number of jobs. Conventional algorithms like Egret Swarm Optimization (ESO) [28], Walrus Optimization Algorithm (WaOA) [29], Darts Game Optimizer (DGO) [26] and Beluga Whale Optimization (BWO) [27] were employed in the analysis process to identify the efficacy of the developed model. The analysis process was effectively accomplished by the five configurations. The configuration details adopted in the HDG-BWO-based job scheduling model with optimal load balancing are elucidated in Table II.

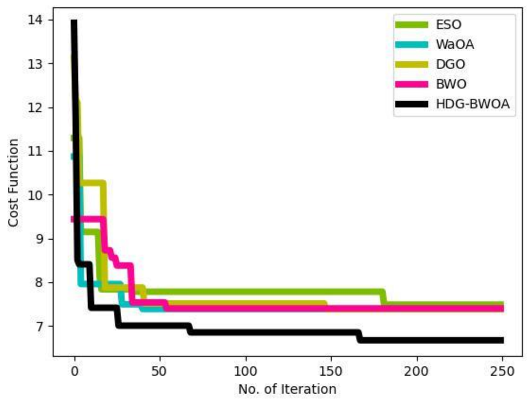
International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

B. Cost function analysis

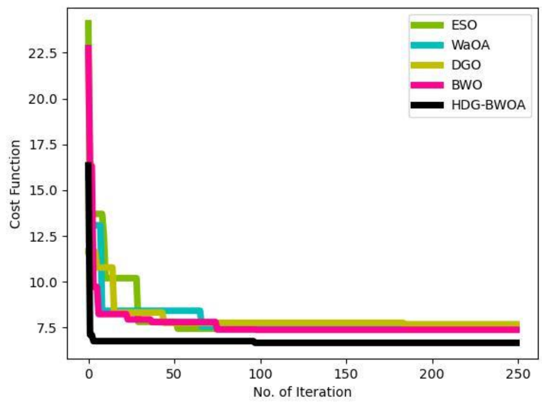
The convergence examination of the explored HDG-BWO-based job scheduling process is specified in Fig. 5. When the number of iterations is raised, the convergence rate of the HDG-BWO-based job scheduling process decreases. When the iteration value is considered as 100, the convergence rate of the HDG-BWO-based job scheduling framework is better than ESO, WaOA, DGO and BWO with 31.66%, 25%, 30% and 26.66%. Thus, the developed HDG-BWO-based job scheduling structure shows better convergence in the job scheduling process.

TABLE II. CONFIGURATION DETAILS OF THE RECOMMENDED HEURISTIC-BASED OPTIMAL JOB SCHEDULING IN A CLOUD ENVIRONMENT

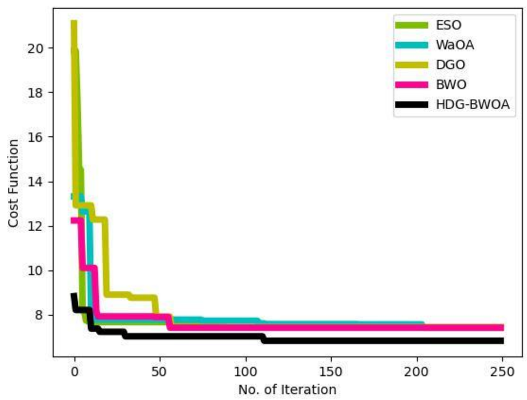
“Configuration”	**Number of servers**	**Memory size**	**CPU**	**Number of jobs**	**Memory size**	**CPU**
1	10	5GB to 7 GB	150 GB to 180 GB	90	950GB to 1.9 GB	25GB to 35GB
2	25	13GB to 18GB	310GB to 340 GB	180	4GB to 8 GB	78GB to 98 GB
3	45	30GB to 34 GB	645GB to 685 GB	285	11GB to 14 GB	118GB to 138GB
4	73	60GB to 65 GB	800GB to 825 GB	385	18 GB to 23GB	148 GB to 150 GB
5	95	78GB to 83 GB	5GB to 7 GB	490	28GB to 32GB	173GB to 178 GB



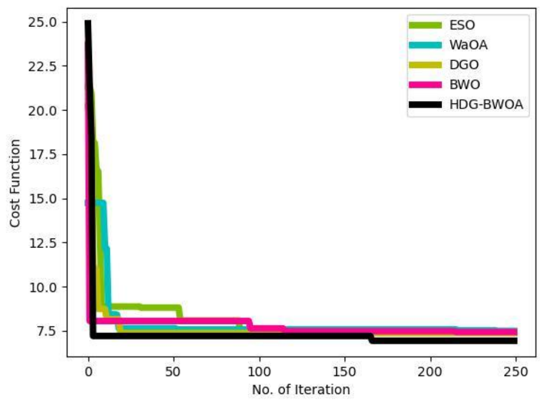
(a)



(b)

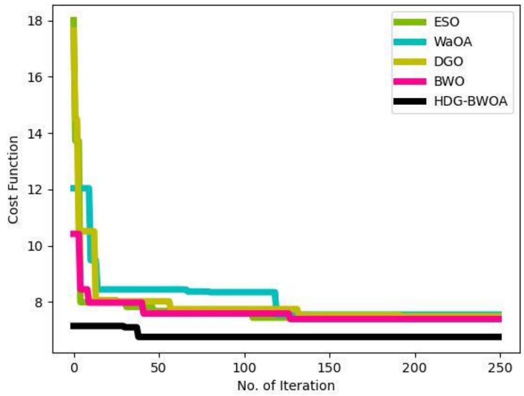


(c)



(d)

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

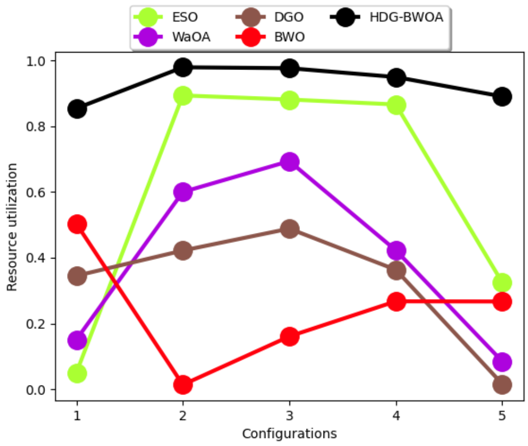


(e)

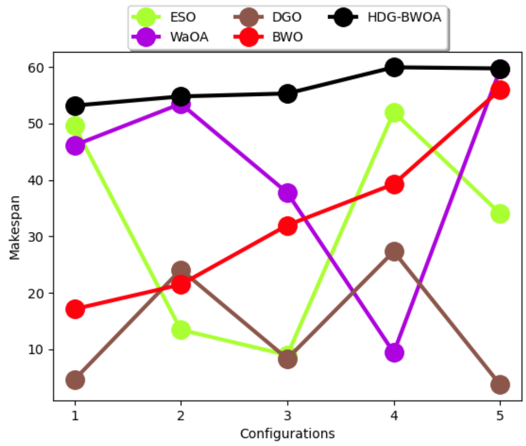
Figure 5. Convergence assessment of the heuristic-based job scheduling process for (a) Configuration 1, (b) Configuration 2, (c) Configuration 3, (d) Configuration 4, (e) Configuration 5

C. Performance evaluation of the suggested model using positive indices among conventional algorithms

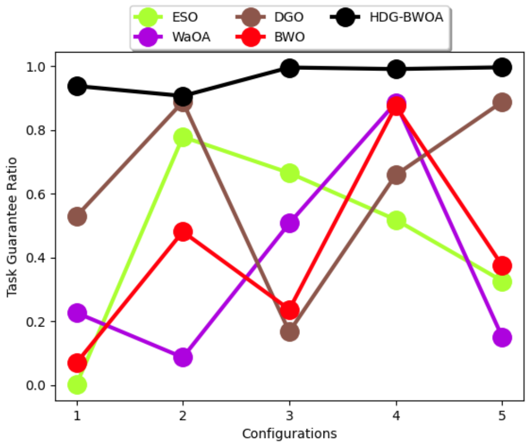
The performance assessment of the explored HDG-BWO-based job scheduling process based on the positive indices is provided in Fig. 6. As mentioned in Fig. 6 (g), the HDG-BWO-based job scheduling process achieves higher throughput than the ESO, DGO, WaOA and BWO with 20%, 18.18%, 9.85% and 23.80% at the configuration 3. Therefore, the throughput rate of the developed HDG-BWO-based job scheduling model is raised than the conventional algorithms.

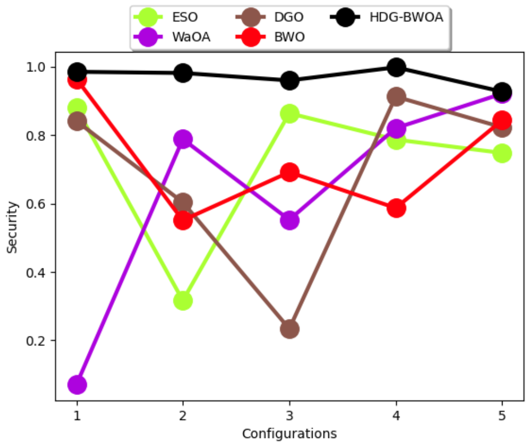


(a)



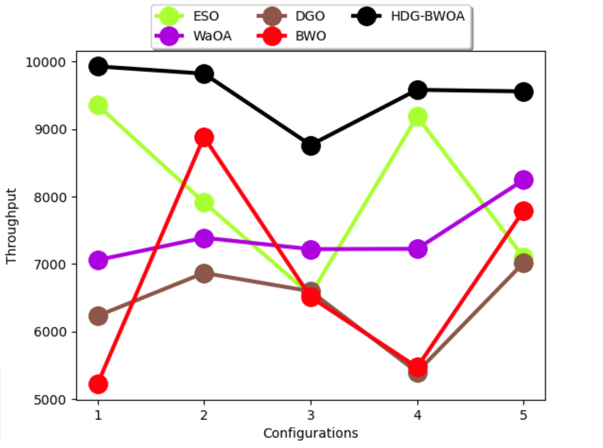
(b)





International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 05 July 2023 **Revised:** 25 August 2023 **Accepted:** 15 September 2023*

(c) (d)

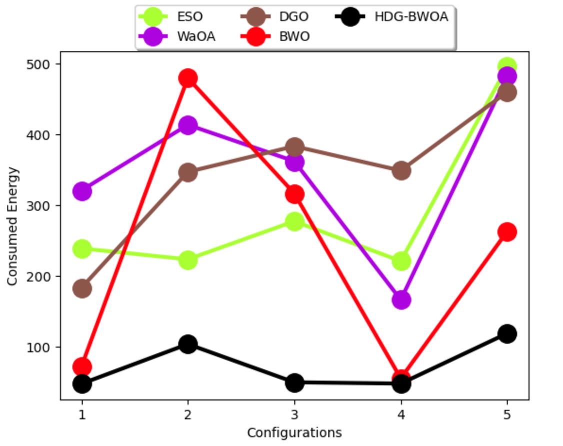


(g)

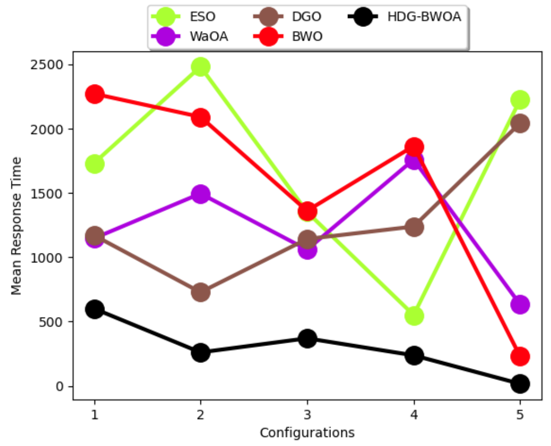
Figure 6. Performance determination of the heuristic-based job scheduling process for (a) Resource utilization, (b) Makespan, (c) Job guarantee ratio, (d) Security, (e) Throughput

D. Performance assessment of the developed model using consumed energy and mean response time

The performance assessment of the developed model on consumed energy and mean response time is designated in Fig. 7. At configuration 2, the mean response time of the developed HDG-BWO-based job scheduling process is shortened than the ESO, DGO, WaOA and BWO with 90%, 68.75%, 83.33% and 88.09%. Similarly, the energy consumption of the developed HDG-BWO-based job scheduling process is very low as compared to the usual algorithm. So, the developed HDG-BWO-based job scheduling process consumes a very low response time than the existing algorithms.



(a)



(b)

Figure 7. Performance evaluation of the heuristic-based job scheduling process for (a) Consumed energy, (b) Mean response time

E. Statistical examination of several algorithms

The statistical examination of the HDG-BWO-based job scheduling model is arranged in Table III. For configuration 1, the best measure of the HDG-BWO-based job scheduling model is upraised with 12.14%, 10.64%, 10.68% and 10.91%. So, the performance of the suggested HDG-BWO-based job allocation model is superior to the existing algorithms.

TABLE III. STATISTICAL EVALUATION OF THE SUGGESTED HEURISTIC-AIDED JOB SCHEDULING MODEL THROUGH SEVERAL ALGORITHMS

ALGORITHMS	ESO [28]	WaOA [29]	DGO [26]	BWO [27]	HDG-BWO
Configuration 1					

BEST	7.485243	7.386536	7.383864	7.398601	6.67225
WORST	11.29473	10.86898	13.11406	9.438971	13.9244
MEAN	7.821605	7.502046	7.717367	7.626955	6.951844
MEDIAN	7.781209	7.386536	7.512571	7.398601	6.851971
STANDARD DEVIATION	0.550523	0.460835	0.852244	0.572806	0.629812
Configuration 2					
BEST	7.441174	7.40804	7.678964	7.376632	6.667063
WORST	24.10892	15.76553	11.65105	22.75028	16.34928
MEAN	7.972295	7.931065	8.0022	7.693664	6.743656
MEDIAN	7.441174	7.55756	7.759791	7.376632	6.667063
STANDARD DEVIATION	1.66709	1.142747	0.804425	1.282697	0.61137
Configuration 3					
BEST	7.422877	7.404899	7.431248	7.406803	6.821539
WORST	19.87506	13.31543	21.10574	12.23289	8.826721
MEAN	7.698397	7.827388	8.038958	7.677048	6.979189
MEDIAN	7.422877	7.570919	7.431248	7.406803	6.821539
STANDARD DEVIATION	1.415384	1.061197	1.601816	0.818233	0.298116
Configuration 4					
BEST	7.413741	7.496861	7.376933	7.436125	6.940097
WORST	21.32683	14.74146	20.20608	23.67385	24.90353
MEAN	8.141291	7.927857	7.623239	7.769633	7.292114
MEDIAN	7.46266	7.583361	7.376933	7.483629	7.210348
STANDARD DEVIATION	1.973923	1.454961	1.427965	1.045217	1.606721
Configuration 5					
BEST	7.422267	7.543718	7.489383	7.389534	6.755631
WORST	18.00856	12.03476	17.64631	10.41462	7.143281
MEAN	7.68976	8.112125	7.887447	7.601801	6.813076
MEDIAN	7.43896	7.543718	7.741969	7.586814	6.755631
STANDARD DEVIATION	0.955224	0.921595	1.039926	0.425251	0.135882

VI. RESULTS AND DISCUSSION

The jobs were distributed to the offered heuristic-based work allocation with an optimal load balancing model to appropriate virtual machines for finishing the job within the limited time period. Here, the user can give the cloud platform a variety of tasks. Sub-jobs were created for this job in order to efficiently complete the job allocation procedure. These sub-jobs were assigned to the virtual machine of the cloud using the HDG-BWO. The HDG-BWO reduces "mean response time, resource utilization, and energy consumption" in the job scheduling process by optimizing the jobs that must be assigned to the virtual resource. The created HDG-BWO-based job planning process had a mean response time at configuration 2 that was lower than that of the ESO, DGO, WaOA, and BWO, which were, respectively, 90%, 68.75%, 83.33%, and 88.09%. Therefore, by utilizing the virtual machine's loading capacity, the generated models efficiently schedule the job to the virtual nodes.

REFERENCES

[1] M. Sharma and R. Garg, "An artificial neural network-based approach for energy-efficient job scheduling in cloud data centers," Sustainable Computing: Informatics and Systems, Vol. 26, June 2020.

[2] A. Marahatta, S. Pirbhulal, F. Zhang, R. M. Parizi, K. K. R. Choo and Z. Liu, "Classification-Based and Energy-Efficient Dynamic Job Scheduling Scheme for Virtualized Cloud Data Center," IEEE Transactions on Cloud Computing, vol. 9, no. 4, pp. 1376-1390, 1 Oct.-Dec. 2021.

[3] H. Yuan, J. Bi and M. Zhou, "Energy-Efficient and QoS-Optimized Adaptive Job Scheduling and Management in Clouds," IEEE Transactions on Automation Science and Engineering, vol. 19, no. 2, pp. 1233-1244, April 2022.

[4] Y. Lu & N. Sun "An effective job scheduling algorithm based on dynamic energy management and efficient resource utilization in green cloud computing environment," Cluster Computing, vol.22, pp. 513–520, 2019.

[5] Mohd Sha Alam Khan & R. Santhosh "Job scheduling in cloud computing using a hybrid optimization algorithm,"Soft Computing, 2021.

[6] J.K.V. Thekkepurayil, David Peter Suseelan& Preetha Mathew Keerikkattil, "Multi-objective Scheduling Policy for Workflow Applications in Cloud Using Hybrid Particle Search and Rescue Algorithm," Service Oriented Computing and Applications, vol. 16, pp. 45–65, 2022.

[7] M. Sharma and R. Garg, "HIGA: Harmony-inspired genetic algorithm for rack-aware energy-efficient job scheduling in cloud data centers," Engineering Science and Technology, an

International Journal, Vol. 23, Issue. 1, pp. 211-224, February 2020.

[8] M.Sohaib Ajmal, Z. Iqbal, F. Z. Khan, M.Ahmad, I. Ahmad, B. B. Gupta, "Hybrid ant genetic algorithm for efficient job scheduling in cloud data centers," Computers and Electrical Engineering, Vol. 95, October 2021.

[9] X. Zhu, L. T. Yang, H. Chen, J. Wang, S. Yin and X. Liu, "Real-Time Jobs Oriented Energy-Aware Scheduling in Virtualized Clouds," IEEE Transactions on Cloud Computing, vol. 2, no. 2, pp. 168-180, April-June 2014.

[10] P. Zhang and M. Zhou, "Dynamic Cloud Job Scheduling Based on a Two-Stage Strategy," IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, April 2018.

[11] Y. Xiong, S. Huang, M. Wu, J. She and K. Jiang, "A Johnson's-Rule-Based Genetic Algorithm for Two-Stage-Job Scheduling Problem in Data-Centers of Cloud Computing," IEEE Transactions on Cloud Computing, vol. 7, no. 3, pp. 597-610, 1 July-Sept. 2019.

[12] A. Ali and M. M. Iqbal, "A Cost and Energy Efficient Job Scheduling Technique to Offload Microservices Based Applications in Mobile Cloud Computing," IEEE Access, vol. 10, pp. 46633-46651, 2022.

[13] B. A. Al-Maytami, P. Fan, A. Hussain, T. Baker and P. Liatsis, "A Job Scheduling Algorithm With Improved Makespan Based on Prediction of Jobs Computation Time algorithm for Cloud Computing," IEEE Access, vol. 7, pp. 160916-160926, 2019.

[14] Z. Chen, J. Hu, X. Chen, J. Hu, X. Zheng and G. Min, "Computation Offloading and Job Scheduling for DNN-Based Applications in Cloud-Edge Computing," IEEE Access, vol. 8, pp. 115537-115547, 2020.

[15] L. Chen, K. Guo, G. Fan, C. Wang and S. Song, "Resource Constrained Profit Optimization Method for Job Scheduling in Edge Cloud," IEEE Access, vol. 8, pp. 118638-118652, 2020.

[16] D. Alsadie, "A Metaheuristic Framework for Dynamic Virtual Machine Allocation With Optimized Job Scheduling in Cloud Data Centers," IEEE Access, vol. 9, pp. 74218-74233, 2021.

[17] S. Pang, W. Li, H. He, Z. Shan and X. Wang, "An EDA-GA Hybrid Algorithm for Multi-Objective Job Scheduling in Cloud Computing," IEEE Access, vol. 7, pp. 146379-146389, 2019.

[18] H. Mahmoud, M. Thabet, M. H. Khafagy and F. A. Omara, "Multiobjective Job Scheduling in Cloud Environment Using Decision Tree Algorithm," IEEE Access, vol. 10, pp. 36140-36151, 2022.

[19] Y. Alahmad, T. Daradkeh and A. Agarwal, "Proactive Failure-Aware Job Scheduling Framework for Cloud Computing," IEEE Access, vol. 9, pp. 106152-106168, 2021.

[20] J. Xu, Z. Hao, R. Zhang and X. Sun, "A Method Based on the Combination of Laxity and Ant Colony System for Cloud-Fog Job Scheduling," IEEE Access, vol. 7, pp. 116218-116226, 2019.

[21] X. Chen, L. Cheng, C. Liu, Q. Liu, J. Liu, Y. Mao, J. Murphy, "A WOA-Based Optimization Approach for Job Scheduling in Cloud Computing Systems," IEEE Systems Journal, vol. 14, no. 3, pp. 3117-3128, Sept. 2020.

[22] D. Alsadie, "TSMGWO: Optimizing Job Schedule Using Multi-Objectives Grey Wolf Optimizer for Cloud Data Centers," IEEE Access, vol. 9, pp. 37707-37725, 2021.

[23] X. Lin, Y. Wang, Q. Xie and M. Pedram, "Job Scheduling with Dynamic Voltage and Frequency Scaling for Energy Minimization in the Mobile Cloud Computing Environment," IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 175-186, 1 March-April 2015.

[24] B. Kruekaew and W. Kimpan, "Multi-Objective Job Scheduling Optimization for Load Balancing in Cloud Computing Environment Using Hybrid Artificial Bee Colony Algorithm With Reinforcement Learning," IEEE Access, vol. 10, pp. 17803-17818, 2022.

[25] Y.Gao, S. Zhang and J. Zhou, "A Hybrid Algorithm for Multi-Objective Scientific Workflow Scheduling in IaaS Cloud," IEEE Access, vol. 7, pp. 125783-125795, 2019.

[26] M. Dehghani, Z. Montazeri, H. Givi, J. M. Guerrero, G. Dhiman, "Darts Game Optimizer: A New Optimization Technique Based on Darts Game," International Journal of Intelligent Engineering and Systems, Vol.13, No.5, 2020.

[27] C. Zhong, G. Li, Z. Meng, "Beluga whale optimization: A novel nature-inspired metaheuristic algorithm," Knowledge-Based Systems, Vol. 251, September 2022.

[28] Z. Chen, A. Francis, S. Li, B. Liao,*, D. Xiao,T. T. Ha, J. Li, L. Ding, and X. Cao, "Egret Swarm Optimization Algorithm: An Evolutionary Computation Approach for Model Free Optimization", Biomimetics, vol. 7(4), pp. 144, 2022.

[29] P. Trojovsky, M.Dehghani, "Walrus Optimization Algorithm: A New Bio-Inspired Metaheuristic Algorithm", Mammalogy, October 2022.