Indonesian Journal of Electrical Engineering and Computer Science
Vol. 38, No. 2, May 2025, pp. 1195~1207
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v38.i2.pp1195-1207

Optimizing cloud tasks scheduling based on the hybridization of darts game hypothesis and beluga whale optimization technique

Manish Chhabra, Rajesh E.
School of Computing Science and Engineering, Galgotias University, Greater Noida, India

Article Info

Article history:

Received Apr 7, 2024
Revised Nov 7, 2024
Accepted Nov 11, 2024

Keywords:

Cloud computing
Metaheuristics
Optimization
Swarm-based optimization
Task scheduling

ABSTRACT

This paper presents the hybridization of two metaheuristic algorithms which belongs to different categories, for optimizing the tasks scheduling in cloud environment. Hybridization of a game-based metaheuristic algorithm namely, darts game optimizer (DGO), with a swarm-based metaheuristic algorithm namely, beluga whale optimization (BWO), yields to the evolution of a new algorithm known as "hybrid darts game hypothesis – beluga whale optimization" (hybrid DGH-BWO) algorithm. Task scheduling optimization in cloud environment is a critical process and is determined as a non-deterministic polynomial (NP)-hard problem. Metaheuristic techniques are high-level optimization algorithms, designed to solve a wide range of complex, optimization problems. In the hybridization of DGO and BWO metaheuristic algorithms, expedition and convergence capabilities of both algorithms are combined together, and this enhances the chances of finding the higher-quality solutions compared to using a single algorithm alone. Other benefits of the proposed algorithm: increased overall efficiency, as "hybrid DGH-BWO" algorithm can exploit the complementary strengths of both DGO and BWO algorithms to converge to optimal solutions more quickly. Wide range of diversity is also introduced in the search space and this helps in avoiding getting trapped in local optima.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Manish Chhabra
School of Computing Science and Engineering, Galgotias University
Greater Noida, India
Email: manishchabra77@gmail.com

1. INTRODUCTION

Recently, cloud technology becomes the foundation for many organizations as well as individuals as it provides a convenient way to access cloud resources like numerous virtual servers, huge virtual storage system, wide area networks (WANs), applications and services remotely through the internet without installing and maintaining them on-premises. Cloud technology three standards are: infrastructure as a service (IaaS), platform as a service (PaaS), software as a service (SaaS) [1]. Five key characteristics are: on-demand self-service, broad network access, resource pooling, rapid elasticity, and measured service [1]. Four deployment models are: public cloud, private cloud, community cloud, hybrid cloud [1].

Key feature of cloud environment is to serve tens of thousands and more of users requests concurrently, which required an efficient task scheduling algorithm [2]. However, in majority, due to improper scheduling, resources are either underutilized or overutilized which increases the cloud resources wastage and thus decline in efficiency. For efficient usage of cloud resources, there are various available scheduling models and optimization criteria.

Numerous classic, deterministic algorithms are available for scheduling user requests. For example, priority scheduling, first-come first-serve (FCFS) scheduling [3], and round robin (RR) scheduling [3] algorithms. But in cloud computing environment, scheduling problem is established as a non-deterministic polynomial time (NP) hard problem [4], [5], so traditional classic scheduling algorithms are unable to solve cloud computing problems. Therefore, various heuristic as well as metaheuristic-based scheduling techniques are widely used in solving cloud optimization problem [6]. There are numerous heuristic and meta-heuristic techniques, but metaheuristic methods are extensively employed in resolving real-time optimization problems. meta-heuristic algorithms are classified into four categories. These are: physics-based algorithms: they rely on the laws of nature, such as black holes, galaxies, and gravitation laws. For example, black hole (BH) algorithm [7], gravitational search algorithm (GSA) [8]. Swarm-based or swarm intelligence algorithms: these are nature inspired, population dependent algorithms. They are established on the interaction between living organisms such as a group of birds, a school of fishes, and a colony of ants. For example, beluga whale optimization (BWO) algorithm [9], ant colony optimization (ACO) [10], particle swarm optimization (PSO) [11]. Evolutionary algorithms: They rely on the process of natural selection. Overall, an evolutionary algorithm contains four steps: initialization, selection, genetic operators and termination. For example, genetic algorithm (GA) [12], differential evolution (DE) [13]. Game-based algorithms: algorithms rely on games uses two strategies. First, modelling the game rules and secondly, the player's different behavior. For example, darts game optimizer (DGO) [14], hide objects game optimization (HOGO) [15].

Problem statement: efficient task scheduling is critical but challenging due to the complexity of cloud environments. Traditional deterministic algorithms, are insufficient because cloud scheduling is a NP-hard problem, leading to the need for metaheuristic approaches. Almost all metaheuristic algorithms are nondeterministic and approximate. These are universal problem-solving algorithms, covering very large scales of problems and generates satisfactory results. Therefore, implementing metaheuristic techniques in cloud computing, it become possible to solve various NP-hard problems in a short duration and hybridization of meta-heuristic algorithms are capable of achieving near optimal solutions in a limited time constraints. Thus, hybridization is considered as an efficient way for solving very complex and sophisticated real-time problems. This paper proposes a "hybrid DGH-BWO" algorithm, which is the hybridization of two metaheuristic algorithms, first game-based algorithm namely DGO [14] and second swarm-based intelligence algorithm namely BWO [9].

The work contribution of hybrid DGH-BWO algorithm includes:

- The proposed algorithm offers more flexibility as it can be easily modified to solve various scheduling problems by modifying the associated algorithms and parameters.
- Scalability is easily achievable as proposed algorithm is capable of handling huge amount of data for processing.
- Sequential hybrid search strategy is used for hybridization of DGO and BWO algorithm which leads to the development of "hybrid DGH-BWO" algorithm.
- A balanced approach is established between expedition and convergence state.
- Introducing a check condition to hybridized both DGO and BWO algorithms strength.
- Efficient task scheduling in cloud environment with maximum resource utilization, task guarantee ratio, and throughput is achievable to an extent.
- In addition to above, minimization of energy consumption and mean response time is also achievable.

In continuation, section 2 presented literature survey. Section 3 describes the proposed method, its analytical model, algorithm, flowchart, and objective function. Section 4 defines the experimental setup. Results and discussion are discussed in section 5. Finally, conclusion along with future aspects is encapsulated in section 6.

2. LITERATURE SURVEY

Kalra and Singh [16], gave a review of five different metaheuristic scheduling techniques. These are ACO, PSO, BAT algorithm, GA, and league championship algorithm (LCA). Authors also defined the optimization criteria to be considered while scheduling tasks in cloud environment, such as makespan, and waiting time. Murad *et al.* [17] gave an overall view of various job scheduling techniques (JST) and resource allocations (RA) techniques for cloud computing environment. Authors classified scheduling techniques as heuristic, metaheuristic and hybrid scheduling. Mohammadzadeh *et al.* [18], gives an overview of various whale optimization algorithm (WOA) variants used by several authors for efficient scheduling in cloud environment. These various task scheduling models are: standard WOA, multi-objective WOA, improved WOA, and hybrid WOA. Main scheduling objectives to achieved are: makespan, budget, quality of service (QoS), energy efficiency, cost, resource utilization, load balancing, performance, efficiency, deadline, and security. Chen *et al.* [19], proposed a method for task scheduling in cloud computing using WOA-based optimization. This proposed method is improved WOA for cloud (IWC) task scheduling. The objective of IWC methodology is to minimize the execution time, load, and cost of the cloud computing system. Zhong *et al.* [9], describes a swarm-based metaheuristic algorithm namely, BWO algorithm for solving various optimization problems. The algorithm is designed on the basis of beluga whales' behavior in the Artic Ocean, which includes swimming in group, looking for prey and finally plunge into the ocean bed i.e. whale drops. Dehghani *et al.* [14], proposed a game-based optimization method, namely DGO. The architecture of DGO is established by replicating the Dart game rules. The key points of DGO algorithm are its simple equations and absence of control parameters. Chen *et al.* [20], proposed a metaheuristic algorithm – egret swarm optimization algorithm (ESOA) – to enhance the balancing between expedition and convergence states of algorithm. The algorithm is stimulated by the hunting skills of two egret species' – the Snowy egret's sit-and-wait approach, and the Great egret's aggressive approach. The discriminant situation is used to establish a balance between two approaches. Trojovsky and Dehghani [21], proposed a biology-stimulated metaheuristic technique known as walrus optimization algorithm (WaOA). The algorithm architecture is based on the natural behaviors of walrus, which include feeding or nourishing singles, migrations, fights with predators or escaping them.

Shanay and Raheem [22], describes artificial bee colony (ABC) algorithm and bee colony optimization (BCO) algorithm for solving travelling salesman combinatorial problems. Saber *et al.* [23], gave outline of various metaheuristic algorithms and its applications in engineering field. For example, GA, is widely used in circuit designing and machine learning. ACO, is employed in routing and scheduling problems, in transportation planning, and telecommunications. PSO, used in power system optimization, robotics, and image processing. Rezk *et al.* [24], describes various metaheuristic techniques employed for solving real-time electrical and civil engineering applications, such as electric vehicle charging scheduling problem, structural design optimization problem. Mostafa and Alsalman [25], uses dolphin swarm algorithm for solving real-time software project scheduling problem.

3. METHOD

Every metaheuristic algorithm mainly consists of two states: expedition and convergence. For finding the global optima, it is difficult to established a balance between expedition and convergence state. Expedition state focus on the global search area and convergence state focus on the local search area. So, in the "hybrid DGH-BWO" algorithm, darts game hypothesis (DGH), section emphasizes expedition by mimicking the throwing of darts towards the target in the search space which marks the boundaries for the expedition. Whereas BWO, section emphasizes convergence by using its echolocation technique. By combining both strategies, hybrid DGH-BWO achieves a balance between expedition and convergence, allowing it to efficiently navigate through the solution space while exploiting promising regions.

3.1. Darts game hypothesis

DGH is based on the concept of DGO. Darts, a competitive shooting sport in which two or more players throw darts at a dartboard. Darts are the small sharp-pointed projectiles and dartboard is a circular shaped target having numerous concentric rings and is divided into 20 radial sections, each section having different assigned points, as shown in Figure 1. Each section is further sub-divided into other sections - single, double and triple scoring sections - by concentric metal wire rings. At the center of the dartboard there are two circles known as inner bull and outer bull, having color red and green respectively.

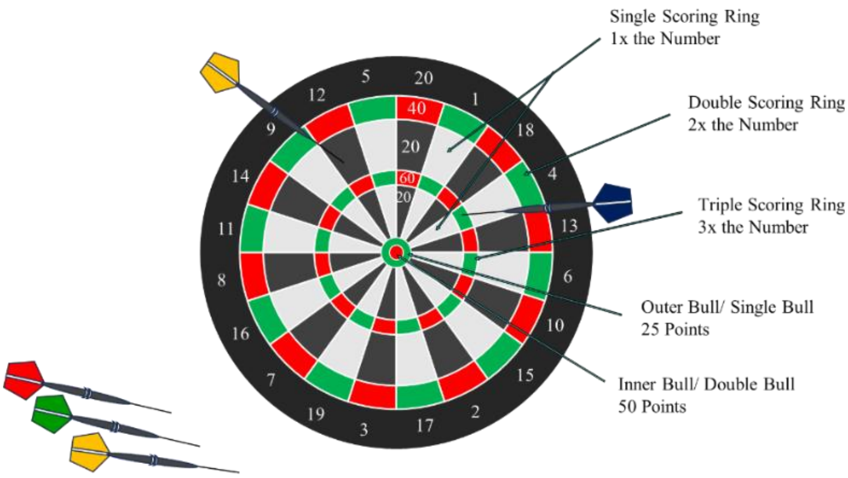


Figure 1. Darts, dartboard and score distribution on dartboard

3.2. BWO

A swarm intelligence, metaheuristic optimization algorithm. The BWO algorithm is executed in three states, which are, expedition, convergence, and whale drops.

3.3. Developing "hybrid DGH-BWO" model

The DGH section of the proposed "Hybrid DGH-BWO" algorithm is inspired by the concept of game of darts. The darts themselves represents the individuals or potential solutions thrown by the players. Players are only the agents that are throwing the darts. Their specific positions or actions are not relevant to the algorithm's calculations. The BWO section of the proposed algorithm is inspired by the behavior of beluga whales' population. In the BWO population, each beluga whale themselves represents the individuals or potential solutions. Therefore, in the "hybrid DGH-BWO" algorithm, word "explorers" represent the analogy of players throwing the darts, and beluga whales. The terminology "explorer" serves as a conceptual framework to understand how the proposed algorithm works.

3.3.1. Developing hybridized phase

Methodology: sequential hybrid search strategy is used for the hybridization of DGO and BWO algorithm which leads to the development of "hybrid DGH-BWO" algorithm. In sequential hybrid search strategy, the phase-based approach is used for achieving hybridization, where the algorithm can switch between DGO and BWO phases based on predefined conditions (check conditions). In this approach, DGO is employed for initial expedition and BWO is employed for fine-tuning solutions.

3.3.2. Procedure of hybridization

Initial expedition phase (DGO):

- Initial expedition is done by the concept of DGO algorithm, where expedition is guided by the concepts of random darts (also named as individuals or players), which are randomly thrown towards the targets (optimal solution) in the search space.
- Target-oriented adjustment is performed in which the individuals' positions are adjusted towards the current best solution by using DGO technique.

Convergence phase (BWO):

- Expedition rate of BWO is constantly reduced, thus focusing more on convergence phase.
- BWO's echolocation mechanism is used to refine the solutions.

3.3.3. Defining the check conditions

The check condition for "Hybrid DGH-BWO" algorithm is determined as (1).

$$Check\ Condition = \begin{pmatrix} \begin{matrix} (1)\ current\ repetition\ Z\ is\ divisible\ by\ any \\ two\ odd\ prime\ numbers, A\ \&\ B. and \end{matrix} \\ (2)\ multiplication\ of\ both\ numbers \\ must\ be\ less\ than\ Z_{max} \end{pmatrix} \tag{1}$$

where Z_{max} , represents the total number of repetitions.

3.4. Mathematical model of hybrid DGH-BWO

A matrix of explorers $E(n \times d)$, where 'n' represents explorers' population size and 'd' represents dimensional position vectors, is represented as (2).

$$E = \left(\begin{array}{c|cccc} E_1 & e_{1,1} & e_{1,2} & \cdots & e_{1,d} \\ E_2 & e_{2,1} & e_{2,2} & \cdots & e_{2,d} \\ \vdots & \vdots & \vdots & e_{g,h} & \vdots \\ E_n & e_{n,1} & e_{n,2} & \cdots & e_{n,d} \end{array}\right) \quad where\ 1 \le g \le n\ and\ 1 \le h \le d \tag{2}$$

Above, $e_{g,h}$ represents the $'g^{th\prime}$ explorer at $'h^{th\prime}$,dimensional location, where $1 \le g \le n\ and\ 1 \le h \le d$. The fitness values related to each explorer is stored in the form of matrix, E_{FitFun} and is represented as (3).

$$E_{FitFun} = \begin{pmatrix} fun(E_1 \to e_{1,1}, \cdots, e_{1,3}, \dots, e_{1,h}, \cdots, e_{1,d}) \\ fun(E_2 \to e_{2,1}, \cdots, e_{2,3}, \dots, e_{2,h}, \dots, e_{2,d}) \\ \vdots \\ fun(E_g \to e_{g,1}, \cdots, e_{g,3}, \cdots, e_{g,h}, \cdots, e_{g,d}) \\ \vdots \\ fun(E_n \to e_{n,1}, \cdots, e_{n,3}, \dots, e_{n,h}, \dots, e_{n,d}) \end{pmatrix} \tag{3}$$

3.4.1. Expedition phase (DGH)

Initial, expedition phase is done by the DGH section of "hybrid DGH-BWO" algorithm, by assigning the explorer E_g values to the fitness function, best and worst fitness function value (Fit_{Best} and Fit_{Worst}) and best and worst variable's values (E_{Best} and E_{Worst}) are established (4)-(7).

$$Fit_{Best} = min(fit)_{n\times1} \tag{4}$$

$$Fit_{Worst} = max(fit)_{n\times1} \tag{5}$$

$$E_{Best} = E(location\ of\ \min(fit), 1{:}\, d) \tag{6}$$

$$E_{Worst} = E(location\ of\ \max(fit), 1{:}\, d) \tag{7}$$

Fitness function normalize value, Fit_g^{Normal}, and probability function value, $Prob_g$, for each g^{th} explorer is calculated as (8), (9).

$$Fit_g^{Normal} = \frac{fit - Fit_{Worst}}{\sum_{k=1}^{n}(fit_k - Fit_{Worst})} \tag{8}$$

$$Prob_g = \frac{Fit_g^{Normal}}{\max(Fit_g^{Normal})} \tag{9}$$

Since, there is total 82 sectors in a dartboard having different scores. Assume, every explorer can throw only three darts in each turn to build his score matrix SM. Each throw scores can be calculated in (10)-(13).

$$C_g = round\left(82 * \left(1 - Prob_g\right)\right) \quad 1 \le g \le n \tag{10}$$

$$SC_g = \begin{cases} SM\left(1{:}\, C_g\right), rand < Prob_g \\ SM\left(C_g + 1{:}\, 82\right), else \end{cases}, SC_g \text{ means } g^{th} \text{ explorer score candidates.} \tag{11}$$

$$S_g = SC_g(q)\ \&\ 1 \le q \le 82,\ S_g \text{ represents every throw score value.} \tag{12}$$

$$S_g^{Normal} = \frac{\sum_{throw=1}^{3} S_g^{throw}}{180}, S_g^{Normal} \text{ denotes the normalized score value.} \tag{13}$$

So, the new updated state of every explorer is established as (14).

$$E_g = E_g + rand(1, d)X\left(E_{Best} - 3S_g^{Normal}E_g\right) \tag{14}$$

With the advancement of each repetition, the predefined check condition determines the execution of convergence phase, which is conducted by BWO section of "hybrid DGH-BWO" algorithm.

3.4.2. Convergence phase (BWO)

First, balance factor bal_{Factor}, is calculated which determines the switching from expedition to convergence state, and is expressed as (15).

$$bal_{Factor} = bal_0(1 - Z/2Z_{max}) \tag{15}$$

Above, bal_0 is the balance factor which changes randomly between (0,1) at each repetition. $'Z'$ represents the present repetition and Z_{max} represents the maximal repetitions. If $bal_{Factor} > 0.5$, Expedition state take place. If $bal_{Factor} \le 0.5$, convergence state happens. As the repetitions $'Z'$ increases, the bal_{Factor} value reduces from (0, 1) to (0, 0.5).

The explorer drops E_{drop} probability is also calculated, which is used for the refinement of optimal solution and is defined as (16).

$$E_{drop} = 0.1 - 0.05\, Z/Z_{max} \tag{16}$$

The probability of explorer drop reduces from $0.1\ to\ 0.05$, with the advancement of repetitions. Since, majority of expedition is already done by the DGH section of the proposed algorithm, the BWO's expedition phase reduces to an extent and more focus is on convergence phase of BWO section.

a. Expedition state

The swimming behavior of these explorers (beluga whales) defines the expedition state. The paired swimming of explorers is used to determine the new updated positions and is computed as follows:

$$\begin{cases} E_{g,h}^{Z+1} = E_{g,d_h}^{Z} + \left(E_{r,d_1}^{Z} - E_{g,d_h}^{Z}\right)(1 + a_1)\sin(2\pi a_2)\,, h = even \\ E_{g,h}^{Z+1} = E_{g,d_h}^{Z} + \left(E_{r,d_1}^{Z} - E_{g,d_h}^{Z}\right)(1 + a_1)\cos(2\pi a_2)\,, h = odd \end{cases} \quad 1 \le g \le n, r \in n \; and \; 1 \le h \le d \tag{17}$$

E_{g,d_h}^{Z} is the present location of g^{th} explorer at the $(d_h)^{th}$ dimension. Where d_h is the random number selected from $d^{th} - dimension.$ $a_1 and\ a_2$ are arbitrary numbers and their values lies between $(0,1)$. $\sin(2\pi a_1)$ and $\cos\,(2\pi a_2)$ defines the locations of synchronized paired explorers fins towards the ocean surface.

b. Convergence state

This defines the preying behavior of explorers. Arbitrary jump strength J_1 and Levy flight strategy is established to improve the convergency. Consider arbitrary numbers, $a_3\ and\ a_4$ lies between $(0,1)$, J_1 is expressed as (18).

$$J_1 = 2a_4(1 - Z/Z_{max}) \tag{18}$$

Levy flight operation L_F is defined as (19).

$$L_F = 0.05\ \times \frac{x \times \Sigma}{|y|^{1/\mathrm{B}}}\,, \quad where\ x\ and\ y \text{ are arbitrary numbers.} \tag{19}$$

$$\Sigma = \left(\frac{\gamma(1+\mathrm{B}) \times \sin\left(\frac{\Pi \mathrm{B}}{2}\right)}{\gamma\left(\frac{(1+\mathrm{B})}{2}\right) \times \mathrm{B} \times 2^{\frac{(\mathrm{B}-1)}{2}}}\right)^{\frac{1}{\mathrm{B}}}, where \text{ B ‘beta’ default value set to } \mathrm{B} = 1.5 \tag{20}$$

So, the analytical model for convergence state is defined as (21).

$$E_g^{Z+1} = a_3 E_{best}^{Z} - a_4 E_g^{Z} + J_1 . L_F . \left(E_r^{Z} - E_g^{Z}\right) \tag{21}$$

where E_{best}^{Z} represents the best position among explorers. E_r^{Z} represents a random r^{th} explorer.

c. Explorer drops state

Explorers are vulnerable species and some explorers can’t escape attack and perished in the bottomless ocean bed. This is known as “Explorer Drops”. Assumed that the explorer drop stimulates a small change and the population size is almost remained constant. Define the E_{step} ,which is the step size of explorer drop and is expressed as (22).

$$E_{step} = (upper_b - lower_b)\ \exp\,(-J_2\, Z/Z_{max}) \tag{22}$$

Where $upper_b$ and $lower_b$ are the upper and the lower limit of variables. J_2 defines the step factor which is related to explorer drop and population size and is defined as (23).

$$J_2 = 2E_{drop} \times n \tag{23}$$

The explorer drops E_{drop} probability is defined in (16). So, the new updated position is computed as (24).

$$E_g^{Z+1} = a_5 E_g^{Z} - a_6 E_r^{Z} + a_7 E_{step} \tag{24}$$

Where $a_5, a_6\ and\ a_7$ are the arbitrary numbers and their values lies between $(0,1)$.

3.5. Proposed algorithm of hybrid DGH-BWO

Sequential hybridization allows the algorithm to maintain a balance between expedition and convergence phase. Check condition determine the switches between the phases. The “Hybrid DGH-BWO” algorithm can be seen in Algorithm 1.

Algorithm 1. Hybrid DGH-BWO algorithm

```
1:  Define the population size n and maximal count of repetitions Z_max. Initialize current
    repetition value Z=1.
2:  Set the check condition variables value 'A' & 'B' using(1).
```

```
3:  Starting positions of all explorers are arbitrary generated and fitness values are
    established using(2), (3) on the basis of objective function, defined in(25).
4:  While Z≤Z_max, Do
5:     For each explorer E_g, 1≤g≤n, Do
6:     If check condition! =True
7:     Calculate 〚Fit〛_Best, 〚Fit〛_Worst, E_Best and E_Worst using (4), (5), (6), and (7).
8:     Compute the fitness function normalized value 〚Fit〛_g^Normal, and probability
    function 〚Prob〛_g, for explorer E_g, using (8) and (9) respectively.
9:     Calculate the normalized score value S_g^Normal, for explorer E_g, using (10), (11),
    (12), and (13).
10:    Explorer E_g, new state is updated using (14).
11:    Check new location boundaries, compute the fitness values and sorted them to determine
    the optimal solution.
12:    Else             //check condition is True
13:    Calculate 〚bal〛_Factor, using (15), and E_drop, using (16).
14:    If 〚bal〛_Factor>0.5           //then it is Expedition phase
15:    Generate d_h,where 1≤h≤d, randomly from dimension.
16:    Select a random explorer E_r.
17:    Explorer E_g, new location is updated using (17).
18:    ElseIf 〚bal〛_Factor≤0.5         // then it is Convergence phase
19:    Update arbitrary jump strength J_1 using (18) and compute Levy Fight operation using
    (19), (20).
20:    Explorer E_g, new location is updated using (21).
21:    End If.    //balance factor.
22:    Check new location boundaries, compute fitness values and sorted them to determine
    the optimal solution.
23:    If 〚bal〛_Factor<E_drop              //Explorer Drops phase
24:    Update the step factor J_2, using (22) & Compute explorer step size E_step, using
    (23).
25:    Explorer E_g, new location is updated using (24).
26:    Check new location boundaries, compute the fitness values and sorted them to determine
    the optimal solution.
27:    End If.                      //Explorer Drops phase
28:    End If.                                 //check condition
29:    Increment value of 'g', for next explorer E_g.
30:    End For.
31:    Determine the latest best solution.
32:    Z=Z+1.
33: End While.
34: Output the latest best optimal solution.
```

3.6. Flowchart

Flowchart of hybrid DGH-BWO algorithm is shown in Figure 2. Transition from DGH's expedition phase to BWO's refinement phase allows the algorithm to maintain a balanced search. Check condition strategically decide the switch between the phases, preventing lengthen focus on suboptimal areas.

3.7. Objective function

The objective function for cloud task scheduling is modelled as:

$$Min\ F(x) = w1 \times (1 - RU) + w2 \times (1 \div Throughput) + w3 \times (1 - TGR) + w4 \times MRT + w5 \times CE. \quad (25)$$

where, RU is resource utilization, TGR is task guarantee ratio, MRT is mean response time, CE is consumed energy. Above, $w1, w2, w3, w4, and\ w5$, are the non-negative weights. $'x'\ in\ F(x)$, signifies a solution vector representing the task-to-virtual machine assignment.

4. EXPERIMENTAL SETUP

Python platform is used to test the proposed "Hybrid DGH-BWO" task scheduling model. For examination, following assumptions are considered:

1) The explorers count, $n = 10$.
2) Maximal count of repetitions, $Z_{max} = 250$.
3) For check conditions in (1), the two odd prime numbers taken are: 'A'=7 and 'B'=9.
4) The proposed model is compared with DGO [14], ESOA [20], BWO [9], and WaOA [21].
5) Five different configurations are taken for servers' and tasks' to be scheduled, for the cloud environment and is shown in Table 1.

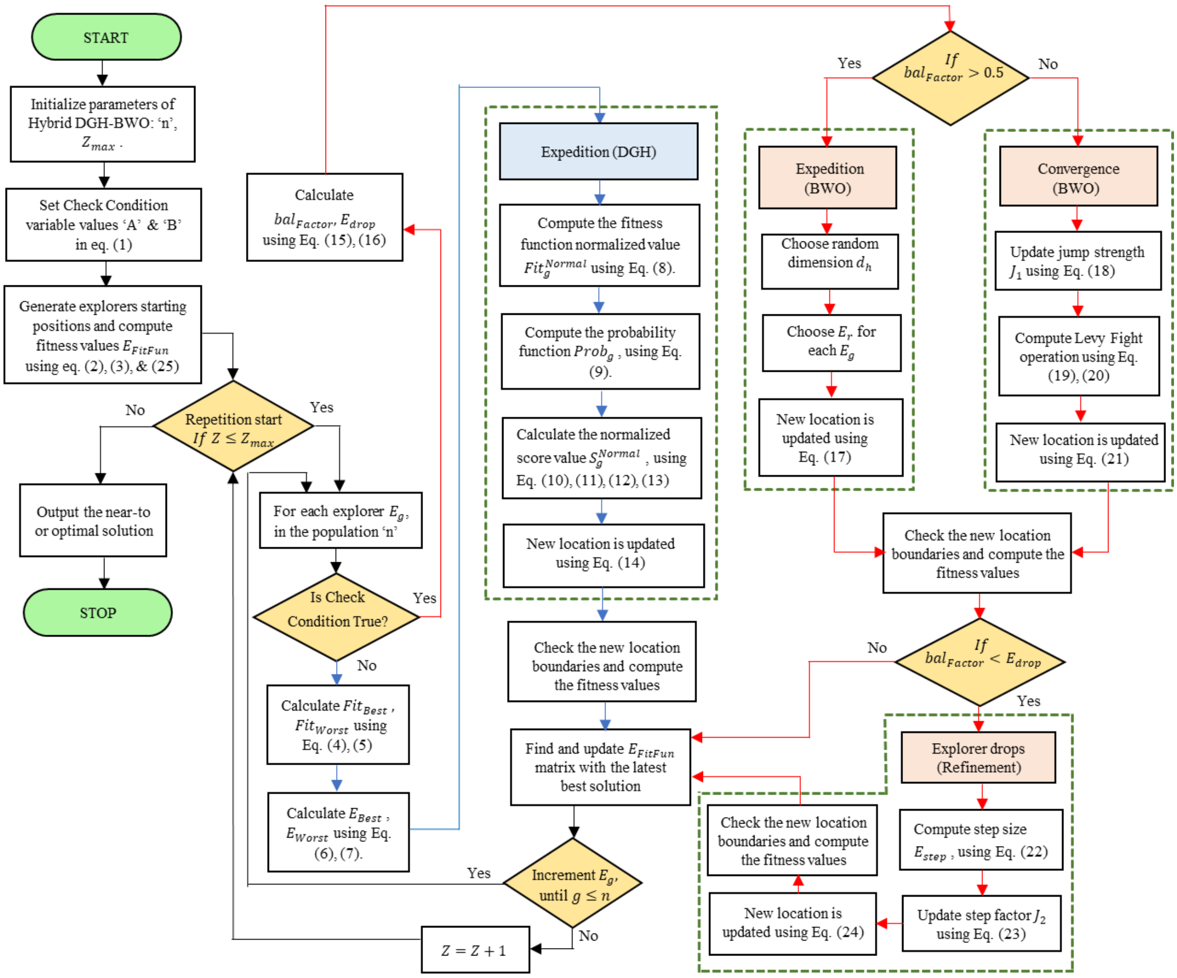


Figure 2. Flowchart of hybrid DGH-BWO algorithm

Table 1. Five different configurations for servers and tasks for the cloud environment

Configuration number	Servers' configuration			Tasks' configuration		
(CN)	Total servers	Memory size	CPU	Total tasks	Memory size	Required CPU
CN1	10	5-7 GB	150-180 GB	90	950 MB-1.9 GB	25-35 GB
CN2	25	13-18 GB	310-340 GB	180	4-8 GB	78-98 GB
CN3	45	30-34 GB	645-685 GB	285	11-14 GB	118-138 GB
CN4	73	60-65 GB	800-825 GB	385	18-23 GB	148-150 GB
CN5	95	78-83 GB	5-7 TB	490	28-32 GB	173-178 GB

5. RESULTS AND DISCUSSION

We performed the cost function, and performance analysis on the basis of resource utilization, task guarantee ratio, security and throughput, and are discussed in subsection 5.1 and 5.2 respectively. Subsection 5.3 shows the performance analysis based on consumed energy and mean response time. In addition, statistical analysis at different configurations, as shown in Table 1, is presented in subsection 5.4. The paper uses graphs to visually compare the algorithms. The graphs show how the "hybrid DGH-BWO" consistently outperforms other algorithms by providing better optimization results across most configurations.

5.1. Cost function examination

The cost function is analyzed at different configurations, CN1, CN2, CN3, CN4, and CN5 and is shown in Figure 3 from Figures 3(a) to 3(e) respectively. The cost function in this paper evaluates and shows how efficiently the proposed "hybrid DGH-BWO" converges as the number of repetitions increases during the task scheduling process, as compared to the other algorithms and thus signifies the algorithm's efficiency in finding optimal solutions.

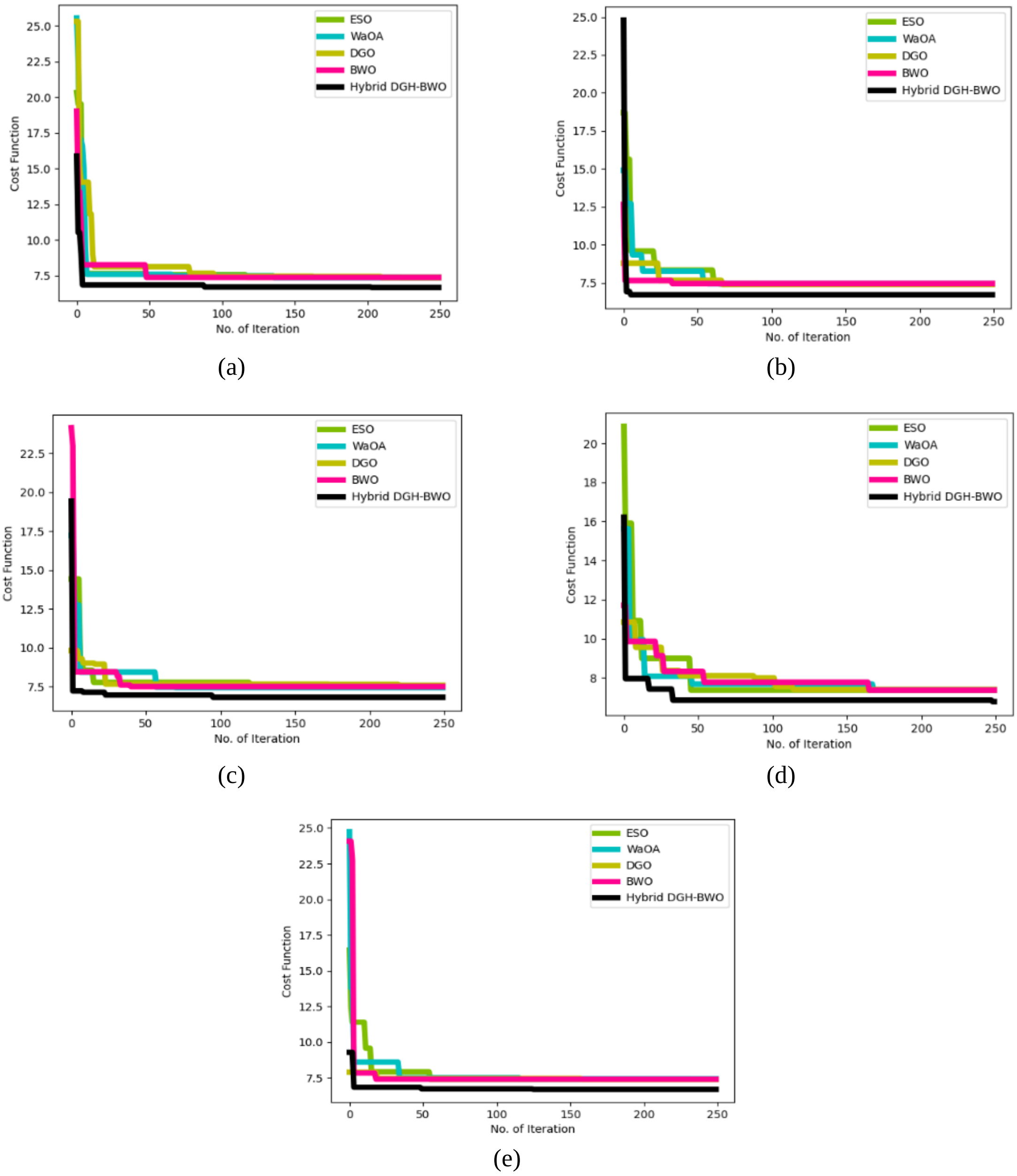


Figure 3. Cost function analysis at configuration (a) CN1, (b) CN2, (c) CN3, (d) CN4, and (e) CN5

5.2. Performance analysis

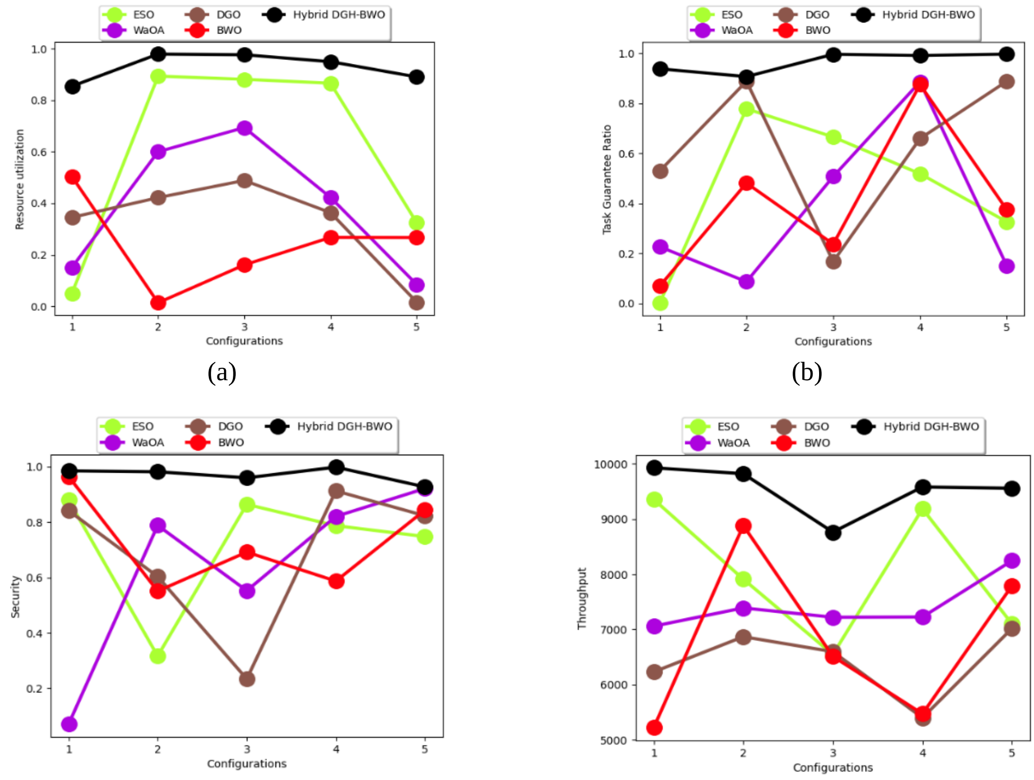
The performance analysis with respect to resource utilization, task guarantee ratio, security and throughput is shown in the Figure 4 from Figures 4(a) to 4(d) respectively. The results shows that the "hybrid DGH-BWO" has better outcomes than other algorithms.

5.2.1. Performance analysis based on consumed energy and mean response time

In Figure 5, Figures 5(a) and 5(b) shows that the "hybrid DGH-BWO" minimizes energy consumption better than others due to its focus on resource optimization, and provides a lower mean response time compared to other algorithms due to its balance of expedition via DGO and convergence via BWO.

5.3. Statistical evaluation

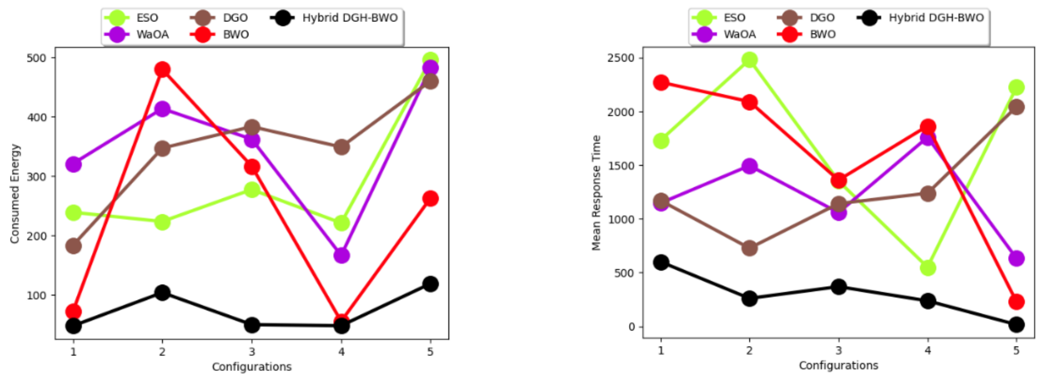
We conducted the statistical analysis at different configurations CN1, CN2, CN3, CN4, and CN5 is shown in Figures 6 to 10 respectively. The results demonstrates that the "hybrid DGH-BWO" algorithm improves efficiency by 10-12% over existing algorithms.



(a) (b)

(c) (d)

Figure 4. Performance analysis at (a) resource utilization, (b) task guarantee ratio, (c) security, and (d) throughput



(a) (b)

Figure 5. Performance analysis with respect to (a) energy consumption and (b) mean response time

Configuration-1Statistical Analysis

ALGORITHMS	ESO	WaOA	DGO	BWO	Hybrid DGH-BWO
BEST	7.36838178300317	7.368491316043557	7.367390187319556	7.374167419913723	6.672665372714016
WORST	20.338483745180696	25.56116057691595	25.321651961627815	19.025675615520562	15.878165683345102
MEAN	7.688328759075522	7.742712524501012	8.003531807634449	7.639796344120479	6.823426657004076
MEDIAN	7.387009280419318	7.494608857608868	7.459567604813335	7.374167419913723	6.704588455300368
STD	1.5505982315343692	1.856537949260468	1.9537392687903488	0.9796390173350812	0.6846525055358023

Figure 6. Statistical analysis at configuration CN1

Configuration-2Statistical Analysis

ALGORITHMS	ESO	WaOA	DGO	BWO	Hybrid DGH-BWO
BEST	7.397674709682301	7.43471697631119	7.38588144467323	7.438284598882658	6.698457044467162
WORST	18.69576094777728	14.897727915408012	8.776650389420183	12.650782591165889	24.791535746248794
MEAN	7.882629099111465	7.767724317249936	7.564063298186129	7.48467314972325	6.788645691023047
MEDIAN	7.397674709682301	7.43471697631119	7.38588144467323	7.438284598882658	6.698457044467162
STD	1.4307252798240728	0.9953368135564922	0.40686397434535476	0.33410192557063073	1.1664602516258928

Figure 7. Statistical analysis at configuration CN2

Configuration-3Statistical Analysis

ALGORITHMS	ESO	WaOA	DGO	BWO	Hybrid DGH-BWO
BEST	7.495833685112027	7.428649992884295	7.585552985763694	7.499288519204497	6.812783557024672
WORST	14.416073581140868	17.217586142189326	9.815240424576414	24.132671689307493	19.426275142778678
MEAN	7.820802287232919	7.799337738922797	7.790258951342176	7.74562639013	6.935507353870513
MEDIAN	7.495833685112027	7.428649992884295	7.659592555093837	7.499288519204497	6.812783557024672
STD	1.0640873538657913	1.1394789871712037	0.4609939693136706	1.4532738636393763	0.7988747739598957

Figure 8. Statistical analysis at configuration CN3

Configuration-4Statistical Analysis

ALGORITHMS	ESO	WaOA	DGO	BWO	Hybrid DGH-BWO
BEST	7.374175924254214	7.3916350847847045	7.399585278647307	7.369833634663218	6.782438695790979
WORST	20.864103343585047	15.632265471746955	10.852764050195757	11.709911010048922	16.2126520721572
MEAN	7.89831491653364	7.856218396906079	7.8943004077531445	7.921457482740196	7.004423058214285
MEDIAN	7.374175924254214	7.681972555265181	7.399585278647307	7.77260549279379	6.860925128911299
STD	1.600016658863219	1.1118832531028817	0.7997059620455418	0.7592487741000197	0.6537671734220168

Figure 9. Statistical analysis at configuration CN4

Configuration-5Statistical Analysis

ALGORITHMS	ESO	WaOA	DGO	BWO	Hybrid DGH-BWO
BEST	7.3944806152867	7.436092825088373	7.383666004845128	7.406705428597218	6.695818064292192
WORST	16.452801849464212	24.734260002420985	7.8995580307341475	24.07808770116044	9.289034325262799
MEAN	7.7154386198418505	7.729155547624336	7.469220936812504	7.627398015114578	6.764039785332119
MEDIAN	7.3944806152867	7.436092825088373	7.467224947954304	7.406705428597218	6.712746794619102
STD	1.0133916590319538	1.2674118787227953	0.12950319635925725	1.7676266301899974	0.28336911012592636

Figure 10. Statistical analysis at configuration CN5

5.4. Findings

5.4.1. Analysis of findings

The study focused to examined the outcomes of hybridization of metaheuristic algorithms – DGO and BWO- which yields to "hybrid DGH-BWO" algorithm and comparing it with other metaheuristic algorithms. Here, we gave some major findings collected from the experimental setup and results. Our findings demonstrate that the outcomes of "hybrid DGH-BWO", is more effective in tasks scheduling over cloud environment. In this study, the graphical views clearly demonstrate that the proposed method leads to higher resource utilization, task guarantee ratio and throughput. In addition, the proposed method also optimized the response time and energy consumption.

5.4.2. Explanation of results

Our study of the following individual algorithms – DGO, BWO, ESO, and WaOA, suggests that DGO and ESO algorithms have a powerful focus on expedition phase, whereas BWO and WaOA algorithms prioritize convergence phase. Moreover, all these algorithms have a tendency of premature convergence, which is a situation in which the algorithm stops searching for a better solution prematurely. This unbalancing between expedition and convergence phase and premature convergence, prevents the algorithms to achieve the optimization criteria of task scheduling over cloud environment. Whereas, our proposed method demonstrates that hybridization of DGO and BWO algorithms magnifies expedition and convergence, which reduces the premature convergence to an extend and this leads towards the superiority of finding better solution. And the outcomes of these individual algorithms and proposed method are graphically visualized in the section 5.1. and section 5.2.

5.4.3. Future aspects

The scope of our findings could be applicable to various real-world applications. Our assumption is assumed to be true, that in the engineering domain, the proposed algorithm could be utilized to optimize the design of bridges, buildings, and other structures by fine-tuning the parameters such as material distribution, load-bearing capacity, and structural integrity. The proposed algorithm could be better utilized in the field of telecommunication to optimize the network routing problem. Moreover, the proposed method could present exciting opportunities for enhancing manufacturing processes across various industries. Manufacturing companies can optimize production schedules, minimize downtime, and enhance resource utilization to improve overall efficiency and productivity in their operations. This leads to reduced lead times, increased throughput, and cost savings in manufacturing processes.

5.4.4. Limitations

This study reported an overall list of challenges to deal with the proposed method. This includes the design complication, where we need to integrate DGO and BWO thoughtfully to prevent unnecessary complexity. Moreover, DGO almost has no control parameters and BWO has some. So, combining them requires adding more parameters, which could make it challenging to find the best configuration. Our analysis further suggests that, finding the right balance between DGO's randomness and BWO's social behavior is very essential, as too much randomness might reduce effectiveness, while overreliance on BWO's social aspects could limit the expedition. We need to be aware of these limitations while considering the future aspects of our work. They offer opportunities for more in-depth research, as well as a unique technique of hybridizing.

6. CONCLUSION

Latest observations suggest that the hybridized metaheuristic algorithms offer more powerful ways of solving optimization problems rather than individual algorithms. Our proposed method "hybrid DGH-BWO" algorithm, holds potential to optimize the task scheduling over cloud environment in an efficient way. In this study, the "hybrid DGH-BWO" algorithm offers significant advantages for cloud task scheduling by combining the strengths of DGO and BWO optimization algorithms. Our results display improved resource utilization, task guarantee ratio and reduced energy consumption and mean response time. These factors increase robustness, enhance scalability and optimize the QoS. Careful designing, proper parameters selection and tuning, and evaluation are essential components for the successful execution of the proposed method.

REFERENCES

[1] P. Mell and T. Grance, *The NIST definition of cloud computing*. NIST SP 800-145, 2011.
[2] T. Aladwani, "Types of task scheduling algorithms in cloud computing environment," in *Scheduling Problems - New Applications and Trends*, IntechOpen, 2020. doi: 10.5772/intechopen.86873.
[3] N. Chowdhury, K. M. Aslam Uddin, S. Afrin, A. Adhikary, and F. Rabbi, "Performance evaluation of various scheduling algorithm based on cloud computing system," *Asian Journal of Research in Computer Science*, pp. 1–6, Oct. 2018, doi: 10.9734/ajrcos/2018/v2i124768.
[4] R. Ravinder and N. Vurukonda, "Cloud computing: an efficient load balancing and scheduling of task method using a hybrid optimization algorithm," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 32, no. 3, pp. 1545-1556–1556, Dec. 2023, doi: 10.11591/ijeecs.v32.i3.pp1545-1556.
[5] A. Subhi Abdalkafor, A. Abdalqahar Jihad, and E. Tariq Allawi, "A cloud computing scheduling and its evolutionary approaches," *Indonesian Journal of Electrical Engineering and Computer Science (IJEECS)*, vol. 21, no. 1, pp. 489–496, Jan. 2021, doi: 10.11591/ijeecs.v21.i1.pp489-496.
[6] M. Abdel-Basset, R. Mohamed, W. A. Elkhalik, M. Sharawi, and K. M. Sallam, "Task scheduling approach in cloud computing environment using hybrid differential evolution," *Mathematics*, vol. 10, no. 21, Oct. 2022, doi: 10.3390/math10214049.
[7] H. Deeb, A. Sarangi, D. Mishra, and S. K. Sarangi, "Improved black hole optimization algorithm for data clustering," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 8, pp. 5020–5029, Sep. 2022, doi: 10.1016/j.jksuci.2020.12.013.
[8] Z. Yang, Y. Cai, and G. Li, "Improved gravitational search algorithm based on adaptive strategies," *Entropy*, vol. 24, no. 12, Dec. 2022, doi: 10.3390/e24121826.

[9] C. Zhong, G. Li, and Z. Meng, "Beluga whale optimization: a novel nature-inspired metaheuristic algorithm," *Knowledge-Based Systems*, vol. 251, Sep. 2022, doi: 10.1016/j.knosys.2022.109215.

[10] F. U. Zambuk, A. Ya'u, M. Jiya, N. Ado, B. Ja'afaru, and A. Muhammad, "Efficient task scheduling in cloud computing using multi-objective hybrid ant colony optimization algorithm for energy efficiency," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 3, 2021, doi: 10.14569/IJACSA.2021.0120353.

[11] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, Aug. 2022, doi: 10.1007/s11831-021-09694-4.

[12] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[13] M. F. Ahmad, N. A. M. Isa, W. H. Lim, and K. M. Ang, "Differential evolution: a recent review based on state-of-the-art works," *Alexandria Engineering Journal*, vol. 61, no. 5, pp. 3831–3872, May 2022, doi: 10.1016/j.aej.2021.09.013.

[14] M. Dehghani, Z. Montazeri, H. Givi, J. Guerrero, and G. Dhiman, "Darts game optimizer: a new optimization technique based on darts game," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 5, pp. 286–294, Oct. 2020, doi: 10.22266/ijies2020.1031.26.

[15] M. Dehghani *et al.*, "HOGO: hide objects game optimization," *International Journal of Intelligent Engineering and Systems*, vol. 13, no. 4, pp. 216–225, Aug. 2020, doi: 10.22266/ijies2020.0831.19.

[16] M. Kalra and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," *Egyptian Informatics Journal*, vol. 16, no. 3, pp. 275–295, Nov. 2015, doi: 10.1016/j.eij.2015.07.001.

[17] S. A. Murad, A. J. M. Muzahid, Z. R. M. Azmi, M. I. Hoque, and M. Kowsher, "A review on job scheduling technique in cloud computing and priority rule based intelligent framework," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 6, pp. 2309–2331, Jun. 2022, doi: 10.1016/j.jksuci.2022.03.027.

[18] A. Mohammadzadeh, A. Chhabra, S. Mirjalili, and A. Faraji, "Use of whale optimization algorithm and its variants for cloud task scheduling: a review," in *Handbook of Whale Optimization Algorithm*, Elsevier, 2024, pp. 47–68. doi: 10.1016/B978-0-32-395365-8.00010-5.

[19] X. Chen *et al.*, "A WOA-based optimization approach for task scheduling in cloud computing systems," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3117–3128, Sep. 2020, doi: 10.1109/JSYST.2019.2960088.

[20] Z. Chen *et al.*, "Egret swarm optimization algorithm: an evolutionary computation approach for model free optimization," *Biomimetics*, vol. 7, no. 4, Sep. 2022, doi: 10.3390/biomimetics7040144.

[21] P. Trojovský and M. Dehghani, "Walrus optimization algorithm: a new bio-inspired metaheuristic algorithm." Oct. 24, 2022. doi: 10.21203/rs.3.rs-2174098/v1.

[22] H. A. Shabat and K. R. Raheem, "Analysis study of the bee algorithms as a mechanism for solving combinatorial problems," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 30, no. 2, pp. 1091–1098, May 2023, doi: 10.11591/ijeecs.v30.i2.pp1091-1098.

[23] M. Saber, A. A. Abdelhamid, and A. Ibrahim, "Metaheuristic optimization review: algorithms and applications," *Journal of Artificial Intelligence and Metaheuristics*, vol. 3, no. 1, pp. 21–30, 2023, doi: 10.54216/JAIM.030102.

[24] H. Rezk, A. G. Olabi, T. Wilberforce, and E. T. Sayed, "Metaheuristic optimization algorithms for real-world electrical and civil engineering application: a review," *Results in Engineering*, vol. 23, Sep. 2024, doi: 10.1016/j.rineng.2024.102437.

[25] B. S. Mostafa and F. A. Alsalman, "Application project task scheduling using dolphin swarm technology," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 1, pp. 549–557, Jul. 2021, doi: 10.11591/ijeecs.v23.i1.pp549-557.

BIOGRAPHIES OF AUTHORS

Manish Chhabra is currently working as part-time research scholar in School of Computing Science and Engineering, Galgotias University, Greater Noida. He has done M.Sc. CS from Jiwaji University Gwalior, Master of Computer Application (MCA) Gold Medalist, from GJU Hisar (Haryana), M.Tech (computer science) from AKTU University, Lucknow, having 15+ years of academic experience in the field of computer science and engineering and has been associated with Galgotias University since 2021. Currently working as assistant professor in AI-ML Department of Vardhaman College of Engineering Hyderabad (An Autonomous Institution, Affiliated to JNTUH and Approved by AICTE Accredited by NAAC with A++ Grade). He is Certified Aviatrix Certified Engineer - Multi-Cloud Network Associate Course. (Serial No. cert_1jx7nxhq) and Several AWS /GCP Cloud /DevOps certification course and learning path from LinkedIn learning platforms. He can be contacted at email: manishchabra77@gmail.com.

Dr. Rajesh E. is currently working as professor cum program chair- B. Tech (Specialization) and M.Tech program in School of Computing Science and Engineering, Galgotias University, Greater Noida. He has 15+ years of multicultural academic experience in the field of computer science and engineering and has been associated with Galgotias University since 2018. Being a passionate teacher, he believes that teaching is not merely restricted to make the students understand the underlying concepts of a course but also to develop critical thinking and evaluate alternate approaches for problem solving. His area of research includes data science, computational intelligence, and networks. He has published more than 30+ research articles in well reputed SCI, Scopus journal and IEEE conferences. He can be contacted at email: rajsugan20@gmail.com.