Hindawi
Mathematical Problems in Engineering
Volume 2021, Article ID 3517145, 14 pages
https://doi.org/10.1155/2021/3517145

Research Article

A Hybrid Approach Based on Grey Wolf and Whale Optimization Algorithms for Solving Cloud Task Scheduling Problem

Jafar Ababneh

Department of Computer Information System and Network/Faculty of Information Technology, The World Islamic Sciences & Education University, Amman, 11947, Jordan

Correspondence should be addressed to Jafar Ababneh; jafar.ababneh@wise.edu.jo

Received 10 May 2021; Revised 6 July 2021; Accepted 14 August 2021; Published 13 September 2021

Academic Editor: Mauro Gaggero

In the context of cloud computing, one problem that is frequently encountered is task scheduling. This problem has two primary implications, which are the planning of tasks on virtual machines and the attenuation of performance. In order to address the problem of task scheduling in cloud computing, requisite nontraditional optimization attitudes to attain the optima of the problem, the present paper puts forth a hybrid multiple-objective approach called hybrid grey wolf and whale optimization (HGWWO) algorithms, that integrates two algorithms, namely, the grey wolf optimizer (GWO) and the whale optimization algorithm (WOA), with the purpose of conjoining the advantages of each algorithm for minimizing costs, energy consumption, and total execution time needed for task implementation, beside that improving the use of resources. Assessment of the aims of the proposed approach is carried out with the help of the tool known as CloudSim. As pointed out by the results of the experimental work undertaken, the proposed approach has the capability of performing at a superior level by comparison to the original algorithms GWO and WOA on their own with regard to costs, energy consumption, makespan, use of resources, and degree of imbalance.

1. Introduction

Among the most well-known uses of metaheuristics for optimization purposes is the scheduling of tasks allocated on sequential or parallel processes. A general definition of task scheduling is the maneuvering of tasks to a particular aggregation of resources distributed in several organizational divisions in timely manner, while tasks refer to atomic units intended for planning and allocation to a resource [1]. Two forms of scheduling are distinguished, namely, static scheduling, in which the scheduler has knowledge of everything regarding tasks and resources, and dynamic scheduling, in which there is no information about the tasks [2]. Scheduling is devised to exploit resources as quickly as possible and the standard algorithms underpinning scheduling include first-come first-served (FCFS), shortest job first (SJF), round robin (RR), Min-Min, and Max-Min.

However, in the context of cloud computing, the abovementioned algorithms are not effective for optimization. Cloud computing is designed to support online service distribution to various destinations and is made up of frontend constituents (e.g., clients and mobile devices) and back-end constituents (e.g., servers, applications, storage devices, and data centers). Cloud computing has become pervasive in every aspect of life owing to the fact that it performs to a high level, is flexible and accessible, ensures a balance of service and demands, and demonstrates high computing power [3–5].

Optimization has importance for every aspect of life and is geared towards improving use of financial resources, energy use, and time, as well as enhancing performance, output, efficiency, and income as much as possible. It can be said that optimization seeks to identify optimal solutions and at the same time exploiting outcomes to the maximum. Optimization is relevant to numerous domains, including engineering, industry, finance, business, projects, management, transport, social sciences, medicine, military and defense, ICT, and numerical problems.

The restricted availability of money, time, and resources influences the practical implementation of optimization [6]. Optimization and modeling based on simulation are primarily concerned with algorithm efficiency, numerical

simulator efficiency and precision, and selection of a suitable algorithm. A variety of optimization methods exist for determining whether a problem is practical or not. Metaheuristics is one such method of exceptional popularity.

Drawing inspiration from nature, metaheuristic algorithms yield reliable outcomes and are more beneficial than standard algorithms [7, 8]. The major classes of metaheuristic algorithms are computational intelligence (CI), biology-based CI (BCI), physics-based CI (PCI), chemistry-based CI (CCI), and mathematics-based CI (MCI) [9]. However, by comparison to other fields, such as physics, chemistry, and mathematics, metaheuristics is generally thought to be yet underdeveloped [10].

Two major categories of metaheuristic optimization algorithms have been identified, namely, local and global. Local algorithms include tabu search (TS), greedy randomized adaptive search procedure (GRASP), and iterated local search (ILS). Meanwhile, global algorithms display an explorative character and include ant colony optimization (ACO), genetic algorithms (GA), and particle swarm optimization (PSO). Metaheuristics have garnered enormous appeal due to the fact that they are uncomplicated, flexible, do not involve mechanisms based on derivation, and avoid local optima [9]. Problems of task scheduling are regarded as optimization problems, whereas a heuristic approach seeks to identify an almost ideal solution with minimal computational time; a metaheuristic approach seeks to identify the most appropriate solution based on a series of rules via number of loops until the best solution is achieved.

From the latest metaheuristic algorithms for addressing optimization problems by identifying global optimal solutions are the grey wolf optimization (GWO) algorithm and the whale optimization algorithm (WOA), which have been respectively introduced in 2014 [11] and 2016 [12] and are inspired by the behavior of grey wolves and whales. These algorithms are primarily characterized by the fact that they extend nominee solutions in every optimization iteration [13].

The problem of cloud task scheduling can be addressed by integrating GWO and WOA in a hybrid approach entitled hybrid grey wolf and whale optimization (HGWWO) algorithm. By integrating the two types of algorithms, hybrid approaches can be created that maximize the strengths of the individual algorithms whilst minimizing their weaknesses [14] that why the need hybridization, beside that hybrid metaheuristic algorithms are more efficient and flexible and perform better than traditional metaheuristic algorithms [15]. Furthermore, hybrid algorithms integrate the features of various metaheuristic approaches and synergy exploitation. A wide range of such algorithms have been developed to promote new-generation metaheuristics.

In [16], the latest approaches for integrating metaheuristics and precise algorithms were explored, while in [17], a novel hybrid approach was proposed, drawing on krill herd and cuckoo search tactics for global optimization tasks. In [18], SA and WOA were integrated in a hybrid approach to facilitate the selection of features, whilst in [19], a hybrid strategy combining GW and differential evolution (DE) mutation was suggested for avoiding local optima. Furthermore, hyperheuristic algorithms have been proposed and summarized in a large number of studies [20–23]. In [24], it was argued that the lowest level of energy of a molecule could be attained by combining GWO and GA, whilst in [25], WOA was made to perform better by employing a DE algorithm within a hyperheuristic algorithm. Optimization can be successfully achieved by exploiting various metaheuristics and integrating them to address diverse problems.

In the present paper, it is proposed that the problem of task scheduling can be solved by implementing a hybrid approach (HGWWO) that integrates GWO and WOA. The remainder of this study is structured in the following way. In Section 2, an overview of cloud task scheduling, GWO, and WOA is provided. In Section 3, other relevant studies are reviewed. In Section 4, the mathematical model is presented, while in Section 5, the proposed process is delineated. In Section 6, the outcomes of the simulation are discussed. Last but not least, in Section 7, conclusions are drawn.

2. Background

Representing a domain of computational intelligence (CI) and metaheuristic algorithms drawing inspiration from the natural world are classified into four types. BCI metaheuristic algorithms include evaluation and swarm algorithms [26]. Swarm algorithms are particularly popular in the majority of fields because it provides greater advantages compared to other algorithms and can address optimization problems more effectively [27, 28].

The latest generation of swarm algorithms using BCI includes WOA and GWO [29]. By contrast to metaheuristic algorithms, hybrid metaheuristic algorithms display a superior performance, being more flexible and effective [15]. As standalone methods, both WOA and GWO present a few limitations, such as being occasionally fixed in local optima solutions [30]. Such limitations can be overcome by applying these algorithms alongside other methods or metaheuristics. On this basis, the present study aims to address the problem of cloud task scheduling by adopting a GWO-WOA hybrid approach.

2.1. The Problem of Task Scheduling. A task is the basic computational unit for running on a node or resources. A job comprises multiple tasks requiring divergent processing capacities and resources. Scheduling can be defined as the assignment of different jobs to existing resources within a certain timeframe according to memory, CPU, nodes, deadline, priority, and other factors. Scheduling has implications for both cloud consumers and cloud providers [31]. However, the cloud environment can be made more efficient and high-performing through resource use optimization. Cloud computing facilitates user access to computing services over the Internet and can be distinguished into three types, namely, public, private, and hybrid clouds [32]. Users, programmers, and administrators access different services offered by cloud providers, namely, Software as

Service (SaaS), Platform as Service (PaaS), and Infrastructure as Service (IaaS), respectively [33]. Task scheduling can be understood as the planning of tasks according to a particular pool of resources that may be allocated between several administrative domains [32].

The design and implementation of scheduling algorithms must take into account several factors, including cost, delay, priority, deadline, security, and energy efficiency, as well as the goals of scheduling. From the perspective of cloud services, the latter can be classified in relation to service providers and service users, as seen in Figure 1 [33]. As multitasking and multiplexing have become essential requirement of the latest applications (e.g., IoT and AI), scheduling algorithms are becoming increasingly in demand [34]. Standard scheduling algorithms (e.g., FCFS, SFJ, RR, Min-Min, and Max-Min) are not very effective at addressing problems of task scheduling and optimization within the context of cloud computing. The complexity and growing frequency of task scheduling in the cloud environment require a metaheuristic approach, whereby the search for the most appropriate solution is based on a series of rules via a number of loops until the best solution is achieved. Such an approach can improve system speed and responsiveness [26].

2.2. Grey Wolf Optimization Algorithm. Drawing on the hunting behavior displayed by wild grey wolves (*Canis lupus*), Mirjalili and colleagues developed the GWO algorithm in 2014 as a method of effectively solving complicated problems based on population agents where living in packs and hierarchical organization comprising several ranks (e.g., α, β, δ, and ω) are some of the characteristics of grey wolves [11]; the hierarchical organization of grey wolves consists of four tiers, with the first tier representing the alpha, decision-maker, and leader (ideal solution), the second tier representing the beta, which supports decision-making and is subordinate to alpha (second best solution), the third tier representing the delta, which is subordinate (second best solution), and the fourth tier representing the omega, which is subordinate (rest of the solutions).

In addition to their hierarchical organization, grey wolves also distinguish themselves through their hunting behavior, which is divided into three stages: prey tracking, chasing, and approaching; prey pursuit, circling, and harassing; and prey attack. For the purposes of GWO development and implementation of optimization hunting method as shown in Figure 2, also the mathematical modeling of grey wolves' hierarchical organization must be undertaken as follows [11]:

(1) Social hierarchy:

α: fittest solution
β: second solution
δ: third solution
ω: rest of the solutions

In the above, the ideal solution (i.e., hunting optimization solution) is led by α, β, and δ, which are followed by ω.

(2) Encircling prey: grey wolves' encircling of prey during hunting is mathematically modeled by the following equations [11]:

$$\vec{D} = |\vec{C}.\vec{X}p(t) - \vec{x}(t)|, \tag{1}$$

$$\vec{X}(t+1) = |\vec{X}p(t) - \vec{A}.\vec{D}|, \tag{2}$$

where t is the current iteration, $\vec{A}$, $\vec{C}$ are coefficient vectors, $\vec{X}p$ is the position vector of the prey, and $\vec{x}$ is the position vector of a grey wolf.

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a}, \tag{3}$$

$$\vec{C} = 2.\vec{r}_2, \tag{4}$$

where $\vec{a}$ is linearly decreased from 2 to 0 and $\vec{r}1$, $\vec{r}2$ is the random vector in range [0, 1].

Hunting: the hunt leader is the alpha, which represents the ideal solution. The first three solutions are retained, while the others are eliminated for position update, as expressed by the following equations [11]:

$$\begin{aligned}
\vec{X}_1 &= \vec{X}_\alpha - \vec{A}_\alpha.\left(\vec{D}_\alpha\right),\\
\vec{X}_2 &= \vec{X}_\beta - \vec{A}_\beta.\left(\vec{D}_\beta\right),\\
\vec{X}_3 &= \vec{X}_\delta - \vec{A}_\delta.\left(\vec{D}_\delta\right),
\end{aligned} \tag{5}$$

$$\begin{aligned}
\vec{D}_\alpha &= \left|\vec{C}_1.\vec{X}_\alpha - \vec{X}\right|,\\
\vec{D}_\beta &= \left|\vec{C}_2.\vec{X}_\beta - \vec{X}\right|,\\
\vec{D}_\delta &= \left|\vec{C}_3.X_\delta - \vec{X}\right|,
\end{aligned} \tag{6}$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{7}$$

(3) Attacking prey: the wolves attack the prey when the latter stops moving.

When the wolves approach the prey, there is a reduction in the random value a $[-a, a]$ from 2 to 0, when the A values are in the range $[-1, 1]$. The attack occurs if $[A]$ is less than 1.

To a certain extent, the encircling strategy reflects exploration. However, a larger number of operators is needed for the purposes of exploration. The GWO element C also prioritizes exploration. Equation (4) shows that the vector includes random values in the range [0, 2] [11].

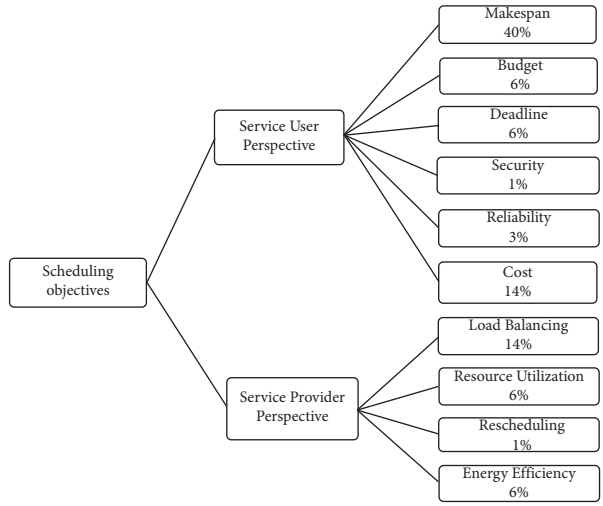

Figure 1: Classification of scheduling algorithms.

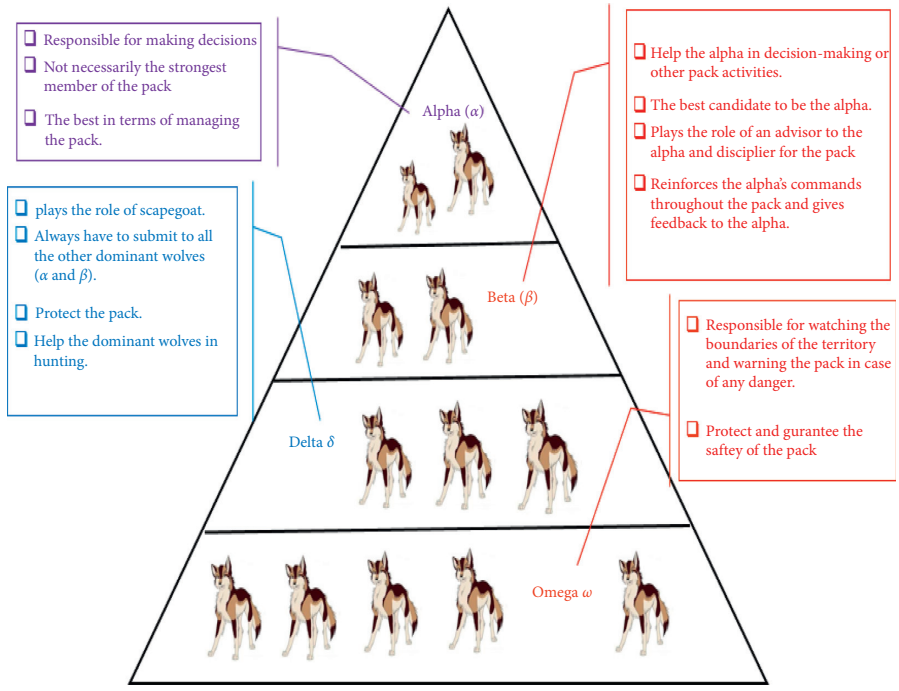

Figure 2: The hierarchical organization of grey wolves [13].

2.3. Wale Optimization Algorithm. Inspired by the bubble-net feeding strategy of humpback whales (*Megaptera novaeangliae*), Mirjalini and Lewis [12] devised the WOA, which is a metaheuristic optimization algorithm based on swarm and CI. The hunting strategy means the whales target prey, dive beneath it, generate bubbles, and move in a spiral shape to direct the prey into the bubble net. Figure 3 shows the hunting strategy [12]. WOA is a highly popular approach with an increasingly wide range of possible applications owing to the numerous benefits it affords, such as the fact that it is efficient, straightforward to apply, and promotes cloud computing robustness and adaptability. The whale behavior of relevance to WOA is prey encircling, spiral bubble-net feeding, and prey search.

(1) Prey encircling: equations (8) and (9) represent whale behavior in terms of prey localization and encircling [12]:

$$\overrightarrow{D} = \left| \overrightarrow{C}.\overrightarrow{X^*}(t) - \overrightarrow{X^*}(t) \right|, \tag{8}$$

$$\overrightarrow{X}(t+1) = \left| \overrightarrow{X^*}(t) - \overrightarrow{A} \cdot \overrightarrow{D} \right|, \tag{9}$$

where t is the iteration steps, $\overrightarrow{A}$. and $\overrightarrow{C}$. are the coefficient vectors, $\overrightarrow{X}$ is the position vector of the best solution, $\overrightarrow{X^*}$ is the position vector, $||$ is the absolute value, (.) element-by-element multiplication. Equations (10) and (11) give $\overrightarrow{A}$. and $\overrightarrow{C}$. [12]:

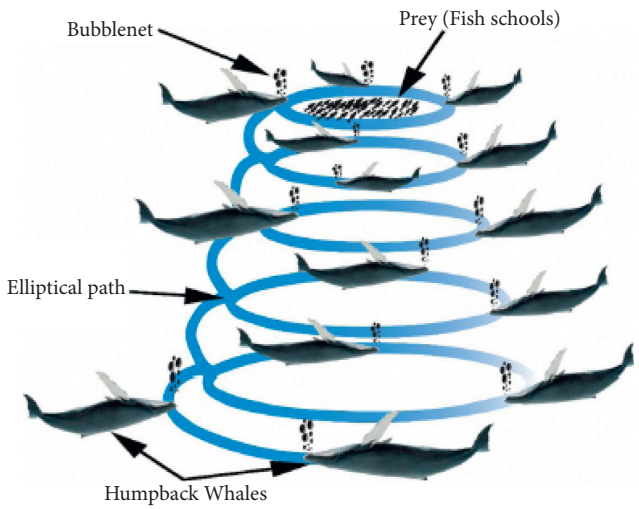

Figure 3: The bubble-net hunting strategy used by humpback whales.

$$\vec{A} = 2\vec{a}.\vec{r}_1 - \vec{a}, \tag{10}$$

$$\vec{C} = 2.\vec{r}_2, \tag{11}$$

where $\vec{A}$ is iteratively (linearly) decreased from 2 down to 0 over the course of iterations and $\vec{r}$ is the random vector with size between [0, 1].

(2) Two methods are used to copy the spiral bubble-net feeding:

An increase in the value of a in the above equation attenuates the encircling method. Calculation of the distance between the whale at (X, Y) and prey at (X, Y) helps to update the spiral shape location.

To model the movement of whales in smaller circles and in a spiral shape, 50% likelihood must be surmised for selection between the method of shrinking encircling and the spiral model for updating whale location. The spiral-shaped movement is given by the following equations [12]:

$$\vec{X}(t+1) = \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \vec{X}(t), \tag{12}$$

$$\vec{X}(t+1) = \begin{cases} \vec{X}*(t) - \vec{A}.\vec{D}, & \text{if } P < 0.5, \\ \overrightarrow{D'}.e^{bl}.\cos(2\pi l) + \vec{X}(t) & \text{if } P \geq 0, \end{cases} \tag{13}$$

where b is a constant for clarifying the shape of logarithmic spiral, l is is the arbitrary number uniformly distributed between [-1, 1], and P is a random number in range [0, 1]

(3) Prey exploration: equations (14) and (15) give the mathematical modeling of prey search, with decision to move in the direction of the location of a random whale rather than the ideal search solution (global optimizer) when {A} > 1 and decision to select ideal agent when {A} < −1 [12].

$$\vec{D} = |\vec{C}.\vec{X}\,\text{rand}(t) - \vec{X}(t)|, \tag{14}$$

$$\vec{X}(t+1) = |\vec{X}\,\text{rand}(t) - \vec{A}.\vec{D}|. \tag{15}$$

3. Literature Review

Within the setting of cloud computing, task scheduling is becoming an increasingly difficult and common problem. This type of problem has been classified by previous studies as an NP-hard problem [35], meaning that it constitutes a heuristic problem. Therefore, numerous algorithms based on heuristics and metaheuristics have been suggested for effectively addressing task scheduling problems by exploiting resource use and reducing makespan as much as possible. Such algorithms include particle swarm optimization (PSO) [36], simulated annealing (SA) [37], genetic algorithms (GA) [38], ant colony optimization (AC) [39], grey wolf optimization (GWO) algorithm [11], whale optimization algorithm (WOA) [12] and hybrid methods; here are some of the recent studies arranged chronically:

In [40], Singh and Bansal proposed a new grey wolf optimizer compound with crossover and opposition-based learning named GWO-XOBL to overcome inadequate variety of wolves prone which led to local optima, and at the end, GWO performance will decrease, the new algorithm evaluated using 13 well-known standard benchmark problems to give expressively performance enhancement compared to GWO and other algorithms.

In [41], Bansal and Singh proposed enhancement to some issues related to GWO algorithm, low exploration and slow convergence rate using explorative equation and opposition-based learning (OBL), where results show better effectiveness than other metaheuristic algorithms; 23 standard benchmark test problems are used to validate the proposed enhancement.

In [42], a modified GWO (MGWO) was proposed to schedule jobs on virtual machines and enhance performance. Outcomes revealed that the algorithm performed better than WOA and GWO in terms of cost and makespan.

In [43], a multiobjective model was proposed to enhance task scheduling in cloud computing via vocalization of WOA, which performed better compared to both WOA and RR owing to makespan, cost, imbalance extent, and resource and energy use.

In [44], prioritization of software project specifications was achieved via GWO, which afforded 30% improvement compared to analytical hierarchy process.

In [13], cloud task scheduling and software specification prioritization were undertaken with a GWO-WOA hybrid method based on the RALIC dataset, yielding precision of 91%.

In [45], various datasets were assessed to address maximum flow from a network source to destination based on GWO and K-means clustering algorithms. Acceptable results were obtained.

In [46], a cloud provider pricing strategy was proposed to cater to providers' requirements, with the overall aim being selection of ideal cloud resources. In this strategy, customers paid more than what they needed to balance the dissonant objectives. Costs were reduced while QoS performance was enhanced by a multiobjective optimization algorithm.

In [47], a model for creation of effective virtual machine allocation and scheduling strategies was proposed, taking into account optimization of data center operational costs and running application performance humiliation. The approach was confirmed to produce resourceful virtual machines and decrease both operational costs and application performance humiliation.

In [48], the profit maximization algorithm (PMA) was proposed to determine price sequential variation in hybrid cloud task for profit increase in private cloud with assurance of service delay bound of delay-tolerant tasks. A secondary problem occurring in every iteration was addressed through a hybrid heuristic optimization algorithm parallel annealing particle swarm optimization, which significantly improved throughput and the profit of private cloud and energy aware scheduling by comparison to five other algorithms.

In [49], an ACO-based multiobjective scheduling approach was suggested. This hybrid approach was designed for scheduling and improving private and public cloud resources according to deadline and cost, number of deadline contraventions, and use of private resources.

In [50], the major cloud problem of energy use was addressed by proposing an aware scheduling algorithm for scientific workflows with specified deadline, which reduced execution time and energy use.

In [51], a deadline and budget distribution-based cost-time optimization algorithm was proposed, taking into account task scheduling deadline and budget.

In [52], PSO coupled with bicriteria priority, deadline, and budget factors was proposed to reduce execution time and cost and performed better than PSO.

In [53], an ACO algorithm was employed to minimize task scheduling makespan, which performed better than FCFS and RR.

In [54], PSO was proposed for undertaking multi-objective cloud task scheduling, which reduced all costs and the transfer and execution time.

In [55], load balancing and makespan minimization were achieved via a load balancing ACO algorithm, which displayed a superior performance to FCFS and ACO.

In [56], GA and AA were integrated in a genetic simulated annealing algorithm, which effectively undertook resource search and distribution based on the QoS requirements of various task types.

In [57], the cost-based algorithm for cloud task scheduling was enhanced through a grouping strategy, which could be useful especially for resources with different costs and computation performance that could lead to task execution completion before the established deadline. The grouping strategy reduced computational costs as well as task execution time.

In [58], particle swarm optimization (PSO) was applied to schedule cloud workflow applications, reducing the overall execution cost three-fold compared to the best-resource selection algorithm.

In [59], task scheduling was successfully achieved based on a genetic algorithm (GA).

4. Mathematical Model

Task scheduler is required when identical resources and virtual machines are simultaneously demanded by customers. Fitness parameters determine the task scheduling method and the ones chosen for this work are resource use, energy, round-trip time latency, and task weight [43]. Equation (16) indicates that the fitness function returns at highest value.

$$\text{Fitness function}\,(F) = \frac{1}{4}\,[\text{FTw} + \text{Ru} + (1 - E) + (1 - L)]. \tag{16}$$

In the above, resource use and required energy for cloudlet application in a virtual machine are respectively denoted by Ru and E, round-trip time latency is denoted by L, and general task weight underpinned by requester category and execution task significance is denoted by FTw. The high priority and latency are emphasized by the $(1-E)$ addition to the equation.

Within cloud computing, the amount of CPU resources and storage required by tasks differs. Equations (17) and (18) give the Ru parameters of CPU cost (C_{cost}) and memory cost (M_{cost}) [60, 61].

$$\text{Ru} = \frac{1}{2}\left[C_{\text{cost}}\,(j) + M_{\text{cost}}\,(j)\right], \tag{17}$$

$$C_{\text{cost}}\,(j) = C_{\text{base}} * C_j * t_{ij} + C_{\text{Trans}}. \tag{18}$$

In the above, the base cost when a resource is used minimally to maintain machine operation to receive incoming requests is denoted by C_{base} and the length of time for Task i (Ti) implementation in virtual machine j (VMj) is denoted by t_{ij}, while VMj cost and the cost related to CPU transmission for target task fulfilment are respectively denoted by C_j and C_{Trans}. M_{cost} is given by the following equation [60, 61]:

$$M_{\text{cost}}\,(j) = M_{\text{base}} * M_j * t_{ij} + M_{\text{Trans}}. \tag{19}$$

Used energy and computational time are correlated [62], so equation (20) defines the energy quantity required by *Ti* in

VM_j as in [63], while equation (21) [63] defines the total energy used by VM_j for cloudlet completion as in [60].

$$E_{\text{cost}}(j) = ECT_{ij} * PO^{j}_{\text{exe}}. \quad (20)$$

In the above, ECT_{ij} and PO^{j}_{exe} respectively denote the energy use time and VM_j power used to execute *Ti*.

$$\text{TE}(j) = \sum_{j=1}^{|\text{VM}|} E_{\text{cost}}(j). \quad (21)$$

The entire phase latency (L) time is denoted by round-trip time, including transmission beginning, reception, and response wait time. Equation (22) gives latency [63].

$$L = \text{EET} + \text{ETT} + 2 * \text{delay}. \quad (22)$$

In the above, EET and ETT respectively denote the anticipated execution time and the anticipated transmission time.

CPU speed, RAM speed, task size, and bandwidth underpin EET. Task implementation is fast for high CPU and RAM speed, meaning that task processing time is inversely correlated with CPU and RAM speed. Furthermore, tasks of short and medium size are executed quicker than long tasks. Equation (23) gives EET, while equation (24) gives ETT [63].

$$\text{EET} = \text{CPU}_{\text{Speed}} + \text{RAM}_{\text{Speed}} + \frac{\text{task size}}{\text{bandwidth}}, \quad (23)$$

$$\text{ETT} = \frac{\text{task size}}{\text{bit rate}}. \quad (24)$$

Task prioritization determines the important factor of FTw, which comprises client type and task weight. Client type comprises several client privilege classes (Table 1) that customers can choose to join and participate in through fee payment and use of requested services provided every week or month according to task size and the volume of reserved resources. Tasks can use existing resources according to their priority weight (high, medium, or low) (Table 1). In the present mathematical model, the employed priority weight factors are 0.4, 0.3, 0.2, and 0.1, which were chosen to give a total of one. Equation (25) gives the FTw.

$$\text{FTw} = \text{CT} * \text{Tw}. \quad (25)$$

In the above, client type and task weight are respectively denoted by CT and Tw.

5. Proposed Algorithm

This paper puts forth a hybrid GWO-WOA algorithm (HGWWO) and the related pseudocode is presented as Algorithm 1. HGWWO was designed for task scheduling into VMs, beginning with an arbitrary group of solutions, with the present solution being deemed the ideal one and serving as the basis for iteration performed until the optimal solution is attained.

The HGWWO stages are

Table 1: The weight, class, and priority of the four tasks.

Classes	Class A	Class B	Class C	Class D
Priority	Urgent	High	Medium	Low
Weight	0.4	0.3	0.2	0.1

(i) Initialization: creation of an arbitrary group of search agents WGi ($i = 1, 2, 3, \ldots, k$)

(ii) Fitness determination: equation (16) is employed to determine the fitness function, with the ideal search agent G$*$ being chosen according to assessment

(iii) Prey encircling: based on the premise of prey location immobility, the search agents encircle the prey and the ideal solution (i.e., prey) is considered to be the present solution; position updating by other search agents is done according to the present optimal agent

(iv) Exploitation: this involves the strategies of shrinking encircling and spiral updating position; shrinking encircling involves determination of new search agent position, with search agent $\vec{r}$ having a value within [−1, 1] and the original position of the search agent and present ideal agent should be used to obtain the new value; the second strategy involves updating agent location based on the spiral technique.

(v) Exploration: updating of search agent location based on a search agent chosen arbitrarily.

(vi) Termination: updating of the value of the ideal search agent is done if a search agent leaves the search area, with commencement of subsequent iteration; this continues until the optimal solution is identified. Briefly, HGWWO is intended to identify the ideal search agent (i.e., cloud task) with the highest fitness value.

This scheduler uses the suggested multiobjective model for cloud task scheduling in the manner of earlier schedulers. Creation of an arbitrary group of search agents is the starting point, followed by task allocation to VMs according to their fitness value when the search agents look for prey. The fittest solution is established as the task with the highest value and is prioritized for allocation to an existing VM based on equation (16), taking into account makespan, resource use, cost, energy consumption, and degree of imbalance (DI).

6. Results and Discussion

A personal computer (Intel Core i-7 processor, 16 GB RAM, and Windows 10 operating system) was used to run CloudSim with Java to empirically assess how the proposed algorithm HGWWO performed. This algorithm was contrasted against WOA and GWO regarding makespan, cost, resource use, energy consumption, and DI.

Makespan and cost respectively refer to the total execution time needed for task implementation [64, 65] and the cost of implementing tasks on particular VMs. Task length, data transfer cost, and storage are determinants of

Input: tasks (T_1, T_2, T_3,..., T_n), virtual machines (V_1, V_2, ..., V_k)
Output: allocated independent tasks to the virtual machines
BEGIN
Initialize the population//initialization phase
Initialize the current best agent WG*
While ($T <$ maximum number of iteration)//optimization phase
Calculate the fitness of each search agent using equation (16)
For each search agent
Update P, R, C, l, and r.
If ($P < 0.5$)
If ($|R| < 1$)
Select WG_α, WG_β, and WG_δ
Update the position of the current search agent using equation (7)
Else if ($|R| > 1$)
Update the position of the search agent according to the randomly selected agent using equation (15).
End if
End if
If ($P >= 0.5$)
Update the position of search agent using equation (12)
End if
End for
If (any search agent goes beyond the search space and amends it)
Calculate the fitness of each search agent
Update WG_α, WG_β, and WG_δ
$T = T + 1$
End while//end main loop
Return WG*

Algorithm 1: Hybrid grey wolf and wale optimization pseudocode.

Table 2: CloudSim parameters.

Entity	Parameters	Values
Cloudlet (task)	Number of cloudlets	100–500
	Length	1000–2000 MIPS
Virtual machine	Number of VMs	8–32
	Bandwidth	1000 Mbps
	Size	10000 MB
	RAM	512 MB
	VMM	Xen
	Operating system	Linux
	Number of CPUs	1
	Policy type	Time shared
	Number of hosts	2
Host	Bandwidth	1000 Mbps
	RAM	2048 MB
	Storage	100000 GB
	Policy type	Time shared
Data center	Number of data centers	2

cost [34, 65]. Cloud service providers benefit from expansion of resource utilization, so permitting customers to rent resources of restricted availability ensures full use of resources and hence profit maximization [34, 66]. Due the energy consumption which is the fourth evaluation Metrix, it is obvious that exploitation for CPU and resources directly possessions power ingestion by tasks, where energy consumption increase if a CPU is not busy also if resources expose for heavy demand [67–69] and the last evaluation metric is DI which calculates the discrepancy through VMs [70], where the following equation is used to calculate it:

$$\text{degree of imbalance}\,(DI) = \frac{T_{\max} - T_{\min}}{T_{\text{average}}}, \tag{26}$$

where $T_{\max}$ is the maximum execution time of VMs, $T_{\min}$ is the minimum execution time of VMs, and T_{average} is the average execution time of VMs.

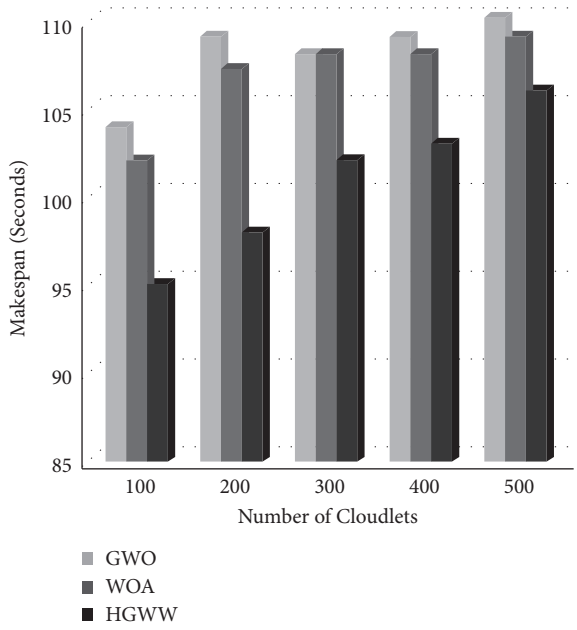

Figure 4: Makespan against number of cloudlets for 8 VMs.

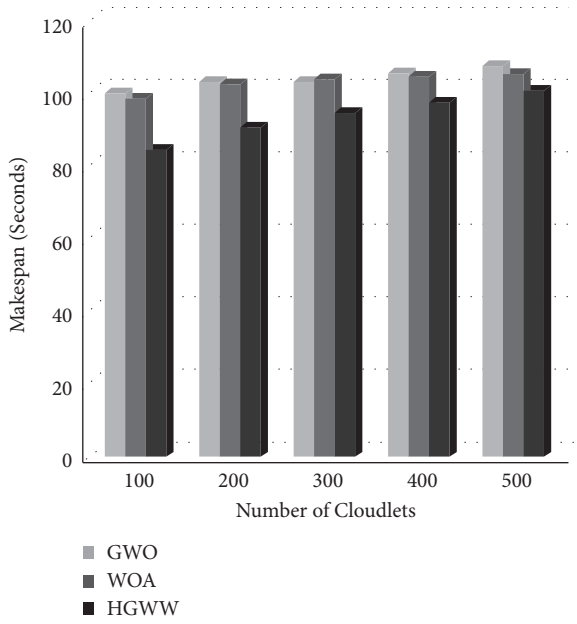

Figure 5: Makespan against number of cloudlets for 16 VMs.

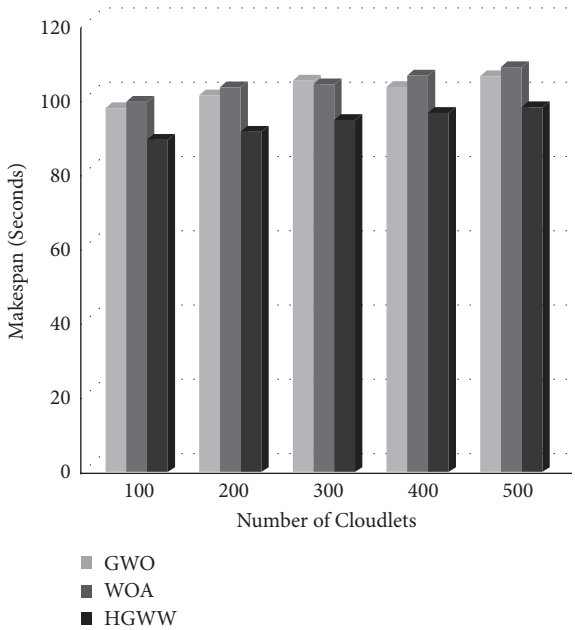

Figure 6: Makespan against number of cloudlets for 32 VMs.

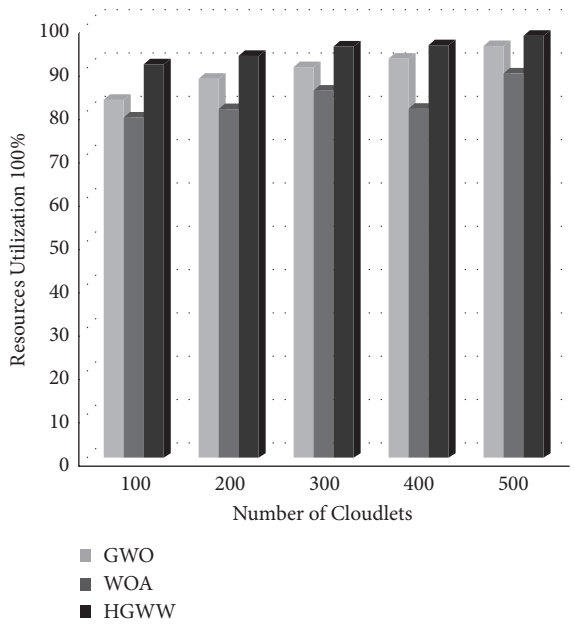

Figure 7: Comparative analysis of the number of cloudlets used by the different algorithms for 16 VMs.

The HGWWO outcomes were contrasted against those of GWO and WOA over a range of 100–500 autonomous cloudlets with 8, 16, and 32 VMs selected in earlier research. Table 2 presents the applied CloudSim configuration, with two hosts, 8, 16, and 32 VMs, two data centers, and 100, 200, 300, 400, and 500 cloudlets, where 30 runs are applied for each scenario then the average is calculated for them.

Regarding makespan, HGWWO performed better compared to traditional GWO and WOA for different number of VMs and cloudlets, Figures 4–6. The superior performance of HGWWO can be attributed to the effective exploration and exploitation capability of this algorithm. As can be seen in Figure 7, HGWWO employed, on average, 8% more resources compared to WOA and 10% more resources compared to GWO for 16 VMs and different number of cloudlets.

The cost of task scheduling associated with different number of cloudlets on 8, 16, and 32 VMs is detailed in Figures 8–10, respectively. Performance as well as the number of appropriate and available resources depend to a significant extent on the number of cloudlets. To put it differently, it is more challenging to attain a global optimal

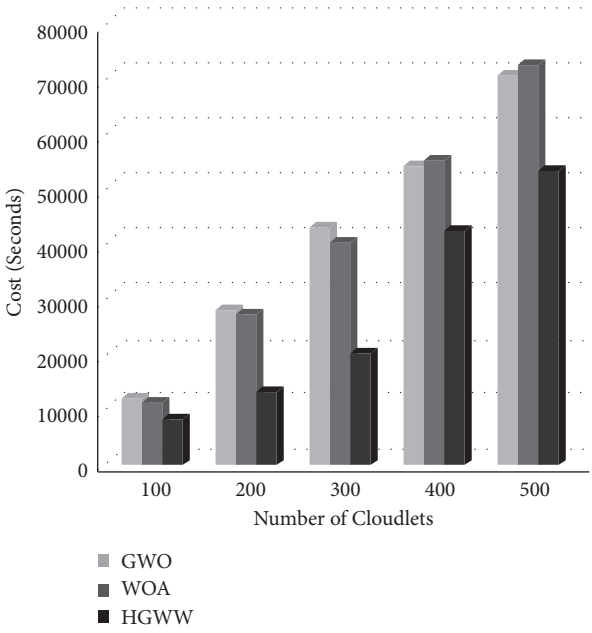

Figure 8: Cost against number of cloudlets for 8 VMs.

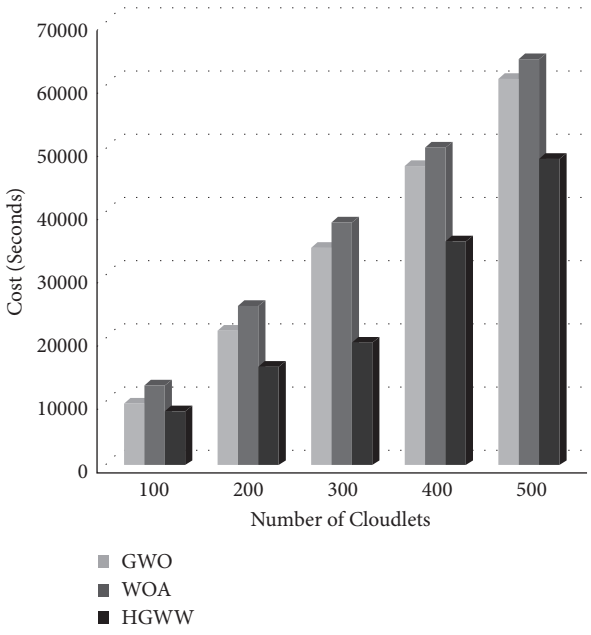

Figure 9: Cost against number of cloudlets for 16 VMs.

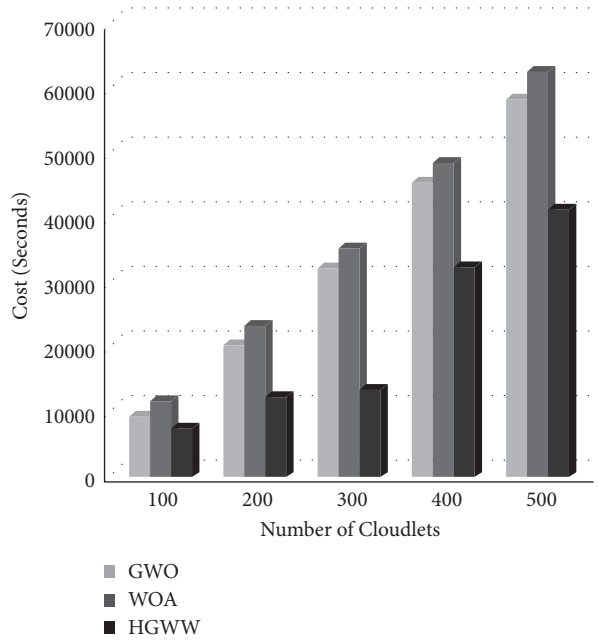

Figure 10: Cost against number of cloudlets for 32 VMs.

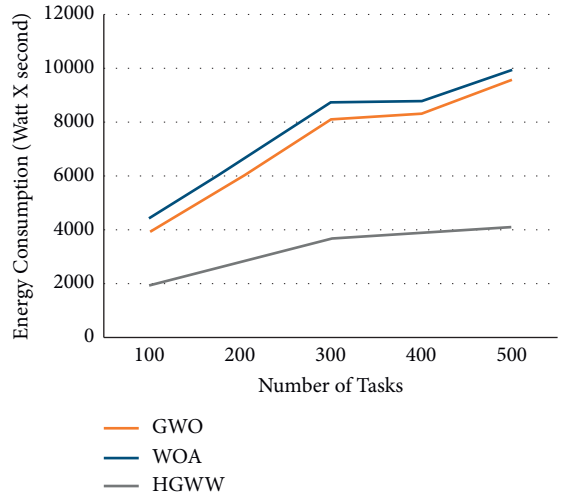

Figure 11: Energy consumption against number of cloudlets for 32 VMs.

solution in cases with an excessive number of tasks. On the other hand, it is straightforward to attain a global optimal solution in cases with excessive amounts of available resources due to the fact that makespan, scheduling cost, and waiting time are all increased when there are numerous tasks but insufficient existing resources. By comparison to WOA, HGWWO achieved a decrease of 40% in the overall execution cost of cloudlet scheduling, whilst by comparison to GWO, HGWWO achieved a 44% decrease in the overall execution cost.

Due the average energy consumption evaluation matrix for HGWWO algorithm of 32 VMs, Figure 11 indicates that HGWWO consumes less energy than WOA and GWO algorithms by 57% and 54.3%, respectively. The overall execution time effects directly on the energy consumption, so HGWWO spends less energy because it has small makespan, but WOA and GWO algorithms have higher makespan, so they spend extra energy.

For tasks (100, 200, 300, 400, and 500) on 32 VMs tested for GWO, WOA, and HGWWO algorithms the DI metric experiment results show more proficient for HGWWO also has a lesser amount of degree of DI regardless of traditional GWO and WOA algorithms which shown in Figure 12.

For calculating percentage of enhancement of HGWWO over WOA and GWO algorithms by using the following equation [71],

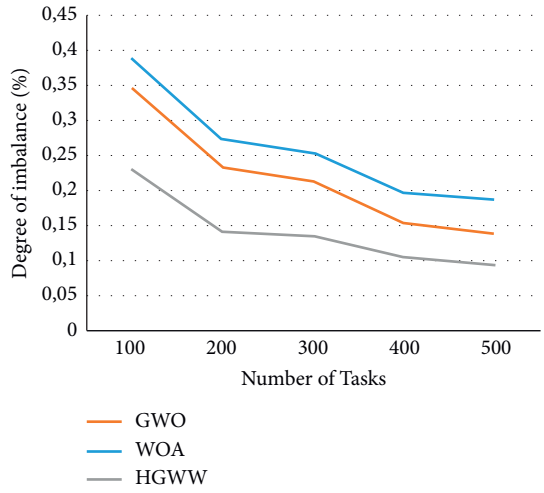

Figure 12: Average degree of imbalance (DI) against number of cloudlets for 32 VMs.

$$\text{Improvement} = \left(1 - \sum \frac{\text{makespan HGWWO}}{\text{makespan WOA or GWO}}\right) * 100. \tag{27}$$

Percentage enhancement is calculated for 32 VMs at 100, 200, 300, 400, and 500 numbers of tasks where result enhancements of HGWWO over WOA algorithm are 11.5%, 12.0%, 10.9%, 11.5%, and 13.0%, respectively, but the result enhancements of HGWWO over GWO algorithm are 9.6%, 9.3%, 12.6%, 8.0%, and 9.7%, respectively, due to the high exploration and exploitation of new proposed HGWWO algorithm, beside that exploration stage shows to the process of looking for the talented provinces of the exploration interplanetary as approximately as probable. In other hand showing the proficiency of local exploration around the talented provinces that attained in the exploration step happened during the exploitation phase. HGWWO achieves enhancement exploration and exploitation aptitude due to the supporters who strengthen globally and randomly the exploration regions for searching.

7. Conclusion

In the context of cloud computing, the problem of task scheduling can be addressed through a wide range of methods of metaheuristic optimization. In the present paper, the GWO and WOA algorithms were integrated into a new algorithm (HGWWO) proposed for solving task scheduling problem. Decreasing makespan, cost, and energy consumption and getting the most out of the resource use are the major goals for the new HGWWO algorithm. The factors considered for evaluating the application of the new algorithm were makespan (total execution time needed for task implementation), cost of implementing tasks on particular VMs, resources use, energy consumption, and degree of imbalance, while CloudSim was the software employed for HGWWO execution. Comparative analysis of HGWWO against the GWO and WOA algorithms in terms of performance revealed that improved results can be achieved with HGWWO than with the other two algorithms; regarding makespan, HGWWO performed better compared to GWO and WOA for different number of VMs and cloudlets because the effectiveness of exploration and exploitation capability of this algorithm also HGWWO has, on average, 8% and 10% more resources compared to WOA and GWO for 16 VMs and different number of cloudlets, respectively; finally, HGWWO has 40% and 44% enhancement in the overall execution cost of cloudlet scheduling compared to WOA and GWO, respectively.

Choosing fitness constraints and their values effect directly on the optimization solution, for insuring parameters values are optimal to guarantee lowest energy depletion and extreme utilization resource, which is the cons of this study. In short coming future, I intend to use evolutionary algorithms such PSO and GA, to solve the task scheduling problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The author declares that he has no conflicts of interest to report regarding the present study.

Acknowledgments

The author expresses his gratitude to the World Islamic Sciences and Education University, Jordan, for administrative and technical support and also to Dr. Raja Almasadeh.

References

[1] B. Sosinsky, "Defining cloud computing," in *Cloud Computing Bible*, pp. 3–22, Wiley Publishing, Inc., Indianapolis, IN, USA, 1st edition, 2011.

[2] T. A. Henzinger, A. V. Singh, V. Singh, T. Wies, and D. Zufferey, "Static scheduling clouds," in *International Conference on Advance Electrical, Electronic, Information, Communication and Bio-Informatices (IEEICB16)*, USENIX Association, Berkeley, CA, USA, 2011.

[3] S. Asadi, M. Nilashi, A. Husin, and E. Yadegaridehkordi, "Customers perspectives on adoption of cloud journal of soft computing and decision support systems 7:4 computing in banking sector," *Information Technology and Management*, vol. 18, no. 4, pp. 305–330, 2020.

[4] F. Mohammed, O. ibrahim, M. Nilashi, and E. Alzurqa, "Cloud computing adoption model for e-government implementation," *Information Development*, vol. 33, no. 3, pp. 303–323, 2017.

[5] E. Yadegaridehkordi, L. Shuib, M. Nilashi, and S. Asadi, "Decision to adopt online collaborative learning tools in higher education: a case of top Malaysian universities," *Education and Information Technologies*, vol. 24, no. 1, pp. 79–102, 2019.

[6] S. Koziel and X. S. Yang, *Computational Optimization, Methods and Algorithms*, Springer, Berlin, Germany, 2011.

[7] X. S. Yang, *Nature-Inspired Metaheuristic Algorithms*, vol. 1st, Luniver Press, London, UK, 2009.

[8] E. G. Talbi, *Metaheuristics: From Design to Implementation*, John Wiley & Sons, Hoboken, NJ, USA, 2008.

[9] H. Kashif, N. Mohd, S. Mohd, C. Shi, and S. Yuhui, "Metaheuristic research: a comprehensive survey," *Artificial Intelligence Review*, vol. 52, pp. 2191–2233, 2019.

[10] M. S. Almufti, "Historical survey on metaheuristics algorithms," *International Journal of Scientific World*, vol. 7, no. 1, pp. 1–12, 2019.

[11] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, vol. 69, no. 4, pp. 46–61, 2014.

[12] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51–67, 2016.

[13] A. Hudaib, R. Masadeh, and A. Alzaqebah, "WGW: a hybrid approach based on whale and grey wolf optimization algorithms for requirements prioritization," *Advances in Systems Science and Applications*, vol. 2, no. 1, pp. 63–83, 2018.

[14] D. T. Pham and T. T. B. Huynh, "An effective combination of genetic algorithms and the variable neighborhood search for solving travelling salesman problem," in *Proceedings of the 2015 Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, pp. 142–149, IEEE, Tainan, Taiwan, November 2015.

[15] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: a survey," *Applied Soft Computing*, vol. 11, no. 6, pp. 4135–4151, 2011.

[16] J. Puchinger and G. R. Raidl, "Combining metaheuristics and exact algorithms in combinatorial optimization: a survey and classification," in *Proceedings of the International Work-Conference on the Interplay between Natural and Arti_cial Computation*, pp. 41–53, Springer, Las Palmas, Canary Islands, Spain, June 2005.

[17] G. G. Wang, A. H. Gandomi, X. S. Yang, and A. H. Alavi, "A new hybrid method based on krill herd and cuckoo search for global optimisation tasks," *International Journal of Bio-Inspired Computation*, vol. 8, no. 5, pp. 286–299, 2016b.

[18] M. M. Mafarja and S. Mirjalili, "Hybrid whale optimization algorithm with simulated annealing for feature selection," *Neurocomputing*, vol. 260, pp. 302–312, 2017.

[19] S. Gupta and K. Deep, "Hybrid grey wolf optimizer with mutation operator," in *Soft Computing for Problem Solving*, pp. 961–968, Springer, New York, NY, USA, 2019.

[20] E. K. Burke, M. Gendreau, M. Hyde et al., "Hyper-heuristics: a survey of the state of the art," *Journal of the Operational Research Society*, vol. 64, no. 12, pp. 1695–1724, 2013.

[21] T. Dokeroglu and A. Cosar, "A novel multistart hyper-heuristic algorithm on the grid for the quadratic assignment problem," *Engineering Applications of Artificial Intelligence*, vol. 52, pp. 10–25, 2016.

[22] M. Beyaz, T. Dokeroglu, and A. Cosar, "Robust hyper-heuristic algorithms for the offline oriented/non-oriented 2D bin packing problems," *Applied Soft Computing*, vol. 36, pp. 236–245, 2015.

[23] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *Journal of Applied Mathematics*, vol. 2013, Article ID 696491, 21 pages, 2013.

[24] M. A. Tawhid and A. F. Ali, "A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function," *Memetic Computing*, vol. 9, no. 4, pp. 347–359, 2017.

[25] M. A. Elaziz and S. Mirjalili, "A hyper-heuristic for improving the initial population of whale optimization algorithm," *Knowledge-Based Systems*, vol. 172, no. 15, pp. 42–63, 2019.

[26] N. Rana, M. S. A. Latiff, S. I. M. Abdulhamid, and H. Chiroma, "Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments," *Neural Computing & Applications*, vol. 32, no. 20, pp. 16245–16277, 2020.

[27] E. Hazir, E. S. Erdinler, and K. H. Koc, "Optimization of CNC cutting parameters using design of experiment (DOE) and desirability function," *Journal of Forestry Research*, vol. 29, no. 5, pp. 1423–1434, 2018.

[28] H. J. Touma, "Study of the economic dispatch problem on IEEE 30-bus system using whale optimization algorithm," *International Journal of Engineering Technology and Sciences (IJETS)*, vol. 5, no. 1, 2016.

[29] T. Dokeroglu1, E. Sevinc, T. Kucukyilmaz1, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering*, vol. 137, 2019.

[30] Y. Ling, Y. Zhou, and Q. Luo, "Lévy flight trajectory-based whale optimization algorithm for global optimization," *IEEE Access*, vol. 5, pp. 6168–6186, 2017.

[31] V. Gopika and K. Vijayanand, "Task scheduling in cloud computing: a survey," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. V, pp. 2258–2266, 2020.

[32] C. Muro, R. Escobedo, L. Spector, and R. P. Coppinger, "Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behavioural Processes*, vol. 88, no. 3, pp. 192–197, 2011.

[33] J. R. Annette, W. A. Banu, and Shriram, "A taxonomy and survey of scheduling algorithms in cloud: based on task dependency," *International Journal of Computer Applications*, vol. 82, no. 15, pp. 20–26, 2013.

[34] P. Singh, M. Dutta, and N. Aggarwal, "A review of task scheduling based on meta-heuristics approach in cloud computing," *Knowledge and Information Systems*, vol. 52, no. 1, pp. 1–51, 2017.

[35] M. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, NY, USA, 1979.

[36] T. Chen, X. Zhang, and Y. Dai, "Task scheduling in grid based on particle swarm optimization," in *Proceedings of the 2006*

Fifth International Symposium on Parallel and Distributed Computing, pp. 238–245, Timisoara, Romania, July 2006.

[37] M. Kashani and M. Jahanshahi, "Using simulated annealing for task scheduling in distributed systems," in *Proceedings of the 2009 International Conference on Computational Intelligence, Modelling and Simulation*, pp. 265–269, Brno, Czech Republic, September 2009.

[38] S. Song, S. K. Hwang, and Y.-K. Kwok, "Risk-resilient heuristics and genetic algorithms for security-assured grid job scheduling," *IEEE Transactions on Computers*, vol. 55, no. 6, pp. 703–719, 2006.

[39] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in *Proceedings of the 2011 Sixth Annual Chinagrid Conference*, pp. 3–9, Liaoning, China, August 2011.

[40] A. Singh and C. Bansal, "Grey wolf optimizer with crossover and opposition-based learning," in *Proceedings of the 6th International Conference on Harmony Search, Soft Computing and Applications ICHSA 2020*, Springer, Istanbul, Turkey, January 2021.

[41] J. C. Bansal and S. Singh, "A better exploration strategy in grey wolf optimizer," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 1, pp. 1099–1118, 2021.

[42] A. Alzaqebah, R. Al-Sayyed, and R. Masadeh, "Task scheduling based on modified grey wolf optimizer in cloud computing environment," in *Proceedings of the 2019 2nd International Conference on New Trends in Computing Sciences (ICTCS)*, pp. 1–6, Amman, Jordan, October 2019.

[43] R. Masadeh, A. Sharieh, and B. Mahafzah, "Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing," *International Journal of Advanced Science and Technology*, vol. 13, no. 3, pp. 121–140, 2019.

[44] R. Masadeh, A. Alzaqebah, A. Hudaib, and A. Rahman, "Grey wolf algorithm for requirements prioritization," *Modern Applied Science*, vol. 12, no. 2, pp. 54–61, 2018.

[45] R. Masadeh, A. Sharieh, and A. Sliet, "Grey wolf optimization applied to the maximum flow problem," *International Journal of Advanced and Applied Sciences*, vol. 4, no. 7, pp. 95–100, 2017.

[46] S. Mireslami, L. Rakai, B. H. Far, and M. Wang, "Simultaneous cost and QoS optimization for cloud resource allocation," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 676–689, 2017.

[47] X. Jin, F. Zhang, L. Wang, S. Hu, B. Zhou, and Z. Liu, "Joint optimization of operational cost and performance interference in cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 697–711, 2017.

[48] Y. Hat, B. Jing, T. Tan, and L. Hu, "Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds," *IEEE Transactions on Automation Science and Engineering*, vol. 14, no. 1, pp. 337–348, 2017.

[49] Z. Lyon, S. Lei Shu, D. Shoubin, C. Yuanfang, and L. Yan, "A multi-objective hybrid cloud resource scheduling method based on deadline and cost constraints," *IEEE Access*, vol. 5, no. 99, pp. 22067–22080, 2016.

[50] Z. Li, J. Ge, H. Hu, W. Song, H. Hu, and B. Luo, "Cost and energy aware scheduling algorithm for scientific workflow with deadline constraint in clouds," *IEEE Transactions on Services Computing*, vol. 11, no. 4, pp. 713–726, 2015.

[51] A. Verma and S. Kaushal, "Deadline constraint heuristic based genetic algorithm for workflow scheduling in cloud," *International Journal of Grid and Utility Computing*, vol. 5, no. 2, 2014.

[52] A. Verma and S. Kaushal, "Bi-criteria priority based particle swarm optimization workflow scheduling algorithm for cloud," in *Proceedings of the 2014 Recent Advances in Engineering and Computational Sciences (RAECS)*, Chandigarh, India, March 2014.

[53] M. Tawfeek, A. EI-Sisi, A. Keshk, and F. A. Torkey, "Cloud task scheduling based on ant colony optimization," in *Proceedings of the 8th International Conference on Computer Engineering & Systems (ICCES)*, pp. 64–69, Cairo, Egypt, November 2013.

[54] L. Guo, G. Shao, and S. Zhao, "Multi-objective task assignment in cloud computing by particle swarm optimization," in *Proceedings of the 2012 8th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Shanghai, China, September 2012.

[55] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in *Proceedings of the 2011 Sixth Annual China Grid Conference*, pp. 3–9, Dalian, China, August 2011.

[56] G. Gan, T. Huang, and S. Gao, "Genetic simulated annealing algorithm for task scheduling based on cloud computing environment," in *Proceedings of the 2010 International Conference on Intelligent Computing and Integrated Systems*, pp. 60–63, Guilin, China, October 2010.

[57] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," in *Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 400–407, Perth, WA, USA, April 2010.

[58] S. Selvarani and G. S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," in *Proceedings of the 2010 IEEE International Conference on Computational Intelligence and Computing Research*, pp. 1–5, Coimbatore, India, December 2010.

[59] C. Zhao, S. Zhang, Q. Liu, J. Xie, and J. Hu, "Independent tasks scheduling based on genetic algorithm in cloud computing," in *Proceedings of the 2009 5th International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1–4, Beijing, China, September 2009.

[60] K. Sreenu and M. Sreelatha, "W-scheduler: whale optimization for task scheduling in cloud computing," *Cluster Computing*, vol. 22, pp. 1087–1098, 2019.

[61] L. Zuo, L. Shu, S. Dong, C. Zhu, and T. Hara, "A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing," *IEEE Access*, vol. 3, pp. 2687–2699, 2015.

[62] D. P. Bunde, "Power-aware scheduling for makespan and flow," *Journal of Scheduling*, vol. 12, no. 5, pp. 489–500, 2009.

[63] M. Sharma and R. Garg, "Energy-aware whale-optmized task scheduler in cloud computing," in *Proceedings of the 2017 International Conference on Intelligent Sustainable Systems (ICISS)*, pp. 121–126, Palladam, India, December 2017.

[64] Q. Wu, D. Yun, X. Lin, Y. Gu, W. Lin, and Y. Liu, "On workflow scheduling for end-to-end performance optimization in distributed network environments," in *Proceedings of the Job Scheduling Strategies for Parallel Processing*, pp. 76–95, 16th International Workshop, JSSPP 2012, Shanghai, China, May 2012.

[65] S. Atiewi, S. Yussof, M. Ezanee, and M. Almiani, "A review energy-efficient task scheduling algorithms in cloud computing," in *Proceedings of the Systems, Applications and Technology Conference (LISAT), 2016*, IEEE Long Island, Farmingdale, NY, USA, April 2016.

[66] A. Al-Shaikh, H. Khatab, A. Sharieh, and A. Sleit, "Resource utilization in cloud computing as an optimization problem," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 6, pp. 336–342, 2016.

[67] X. Wang, Y. Wang, and H. Zhu, "Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm," *Mathematical Problems in Engineering*, vol. 2012, Article ID 589243, 16 pages, 2012.

[68] S. Atiewi, A. Abuhussein, and M. Saleh, "Impact of virtualization on cloud computing energy consumption: empirical study," in *Proceedings of the 2nd International Symposium on Computer Science and Intelligent Control*, p. 24, ACM, Stockholm, Sweden, September 2018.

[69] S. Atiewi, M. Ezanee, S. Yussof, and M. Zalloum, "A power saver scheduling algorithm using DVFS and DNS techniques in cloud computing data centres," *International Journal of Grid and Utility Computing*, vol. 9, no. 4, pp. 385–395, 2018.

[70] N. Almezeini and A. Hafez, "Task scheduling in cloud computing using lion optimization algorithm," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 11, pp. 77–83, 2017.

[71] H. Yin, H. Wu, and J. Zhou, "An improved genetic algorithm with limited iteration for grid scheduling," in *Proceedings of the Sixth International Conference on Grid and Cooperative Computing, 2007 (GCC 2007)*, pp. 221–227, IEEE, Xinjiang, China, August 2007.