VOLUME-13- ISSUE -3 (2019)

Available online at www.ijast.org

International Journal of Advanced Science and Technology

Journal homepage: www.ijast.org

Science and Engineering Research Support Society

Humpback Whale Optimization Algorithm Based on Vocal Behavior for Task Scheduling in Cloud Computing

Raja Masadeh[1], Ahmad Sharieh[2], Basel A. Mahafzah[3]

[1] Computer Science Department, The World Islamic Sciences and Education University, Amman, Jordan

[2] Computer Science Department, The University of Jordan, Amman, Jordan

[3]Computer Science Department, The University of Jordan, Amman, Jordan

[1]raja.masadeh@wise.edu.jo, [2]sharieh@ju.edu.jo, [3]b.mahafzah@ju.edu.jo

A R T I C L E I N F O

Article history:
Publiched May 2019

Keywords:
Optimization, Metaheuristic algorithms, Whale Optimization algorithm, Vocalization Whale Optimization algorithm, Round Robin algorithm, Cloud computing, Independent task scheduling, Multi-objective model, Makespan.

A B S T R A C T

Independent task scheduling is considered one of the most popular issues in cloud computing environment. This study proposes a new metaheuristic optimization algorithm, which is called vocalization of humpback whale optimization algorithm (VWOA). The VWOA mimics the vocalization behavior of humpback whales, and it is employed to improve task scheduling in a cloud computing environment. The VWOA scheduler is based on a suggested multi-objective model. It enables reductions in makespan, cost, and energy consumption and maximizes the utilization of resources. The best optimization solution relies on the fitness parameters and their values, which should remain optimal to ensure minimum energy consumption, maximum resource utilization, and client satisfaction. The experiment results on the tested data showed that the VWOA scheduler has better performance than the results of the traditional whale optimization algorithm (WOA) and round robin (RR) algorithm in terms of makespan, cost, degree of imbalance, resource utilization, and energy consumption. The proposed algorithm has saved 17% and 72% energy compared with WOA and RR algorithms, respectively. The total execution cost of scheduling the tasks using VWOA is decreased by 13% and 22% and the average resource utilization using VWOA is increased by 8% and 35% compared with the WOA and RR algorithms, respectively.

1. Introduction

The needs for huge storage resources have increased in different areas due to the availability of big data. Thus, cloud computing (CC) allows customers to access resources such as software, hardware, and information [1–4]. The CC service provider is responsible for managing the services that are accessed by customers through the Internet [6, 6]. Several types of resources and services are offered by the service provider to the customers. The most important services are Expert as a Service [7], Infrastructure as a Service [8], and Platform as a Service [9]. Several jobs are submitted simultaneously in CC to the available resources. The performance of CC can be ameliorated by assigning available resources to the customers' jobs in an efficient manner.

Corresponding author. Raja Masadeh, Computer Science Department, The World Islamic Sciences and Education, University, Amman, Jordan. E-mail address: raja.masadeh@wise.edu.jo

Task scheduling is one of the most crucial processes of CC, and it plays a significant role of affecting quality of services on the CC [10-12]. Job scheduling in CC aims to select suitable and available resources to execute jobs or to assign computer machines to jobs in such a method that the completion time is reduced as possible. In a task scheduling algorithm, a list of jobs is generated by assigning priority to each job on the basis of different parameters. Jobs are then selected based on their priorities and allocated to available resources that satisfy a predetermined target function.

One of the main goals of the task scheduling process is a reduction in makespan of applications [13, 14].Makespan represents the difference time between starting and ending a sequence of tasks. Thus, algorithms that assign the tasks to the available resources and reduce makespan are needed. Moreover, load balancing is considered a significant goal of task scheduling [15]. Load balancing aims to balance the load of the whole system by transmitting additional tasks from an overloaded virtual machine (VM) to a convenient under loaded VM [16]. In CC, the task scheduling process is known as an NP-complete problem [17]. In this type of problem, the required time to manage the solution varies depending on the size of the problem [18].

Task scheduling optimization algorithms can be classified as metaheuristic and heuristic algorithms. Recently, metaheuristic algorithms have become popular because they are more generic and applicable to solve wide domain issues containing diverse local and global optima than traditional task scheduling algorithms. In other words, metaheuristic algorithms detect the solutions that are optimal or near optimal [19-23]. The heuristic algorithm detects the solution by ignoring some possible paths in the solution space [24-26]. Heuristic algorithms are classified into three classes: list scheduling, cluster scheduling, and task duplication-based scheduling [25, 27].

The list scheduling algorithm is implemented in two phases. In the first phase, a priority value is assigned to each task. In the second phase, based on the priority value, each task is allocated to the available and suitable processor. Thus, list scheduling algorithms work efficiently when the search occurs in a small solution space and then reaches a limited solution set. However, the weakness of this type of heuristic algorithm is its performance, which is heavily dependent on the effectiveness of the heuristics. Thus, a list scheduling algorithm does not show potential in generating consistent outcomes for task scheduling [24, 28].

A clustering algorithm supposes a sufficient and available number of resources to task execution. Moreover, it utilizes as many resources as possible to minimize the schedule’s makespan [21]. The duplication-based algorithm aims to decrease communication delays by performing a key job on more than one processor, thereby minimizing the makespan of the schedule [24, 29].

These algorithms are suitable and applicable for homogeneous systems. In addition, they have unlimited resources [30]. However, metaheuristic algorithms use random choices to find the best solution [25, 30, 31].

The required time for designating resources increases by increasing the number of tasks [5, 30, 32-35]. Therefore, the current algorithms are not suitable for computing with big data. Metaheuristic algorithms permit task scheduling of huge data because they supply enhanced performance in the obscure search space [25], such as the whale optimization algorithm (WOA) [19].The suggested algorithm presented in this paper is based on the vocalization behavior of humpback whales. The main objectives of vocalization are migration and mating. The types of humpback whales for these objectives are singers, adults singing, cow–calf, and the rest of the pod.

The contributions of this paper are summarized as follows. First, a new metaheuristic algorithm is presented, which mimics the leadership hierarchy of vocalization behavior of male humpback whales in nature. Second, a multi-objective task scheduling model for calculating fitness function is proposed to schedule tasks to VMs in a CC environment. Third, a design of the proposed multi-objective task scheduling model based on the vocalization of humpback whale optimization algorithm (VWOA) is introduced.

The paper is organized as follows. Section 2 presents related work, Section 3 presents the proposed multi-objective task schedule model, Section 4 outlines the suggested algorithm (VWOA), and Section 5 describes the WOA. The experimental results and discussion are presented in Section 6. Finally, Section 7 concludes the work.

2. Related Work

Many researchers have presented several modifications of task scheduling in CC and metaheuristic algorithms.

A powerful and enhanced genetic algorithm (GA) is suggested by Keshanchi et al. [32] to optimize the task scheduling solutions. However, the drawback of this algorithm is it takes more time than other algorithms to obtain solutions. Kaur [36] proposed an efficient approach for task scheduling based on GA, which reduces execution cost and time. Their experimental results show that their suggested algorithm provides better performance than other algorithms under heavy load. Lakshmi and Srinivasu [57] proposed a task scheduling algorithm based on GA. This method aims to optimize the waiting time of the system. Their proposed algorithm selects the best series of tasks to be implemented. The main goal of this algorithm is to minimize the waiting time of the available tasks with assistance of the round robin (RR) algorithm. This algorithm exhibits better performance than the traditional RR algorithm.

Sreenu and Sreelatha [25] proposed adaptive multi-objective task scheduling algorithm based on the whale optimization algorithm in CC. The fitness value is calculated by the algorithm. First, the memory and cost function of the central processing unit (CPU) are calculated. Second, the budget cost and makespan of the scheduling process are added. This algorithm assigns the tasks to the VM with a minimum makespan of 7 and minimum average cost of 5.8.Reddy and Kumar [37] suggested multi-objective task scheduling based on WOA. This method considers resource utilization, energy, and quality of service as constraints of the optimization problem. The experimental results showed that the proposed algorithm outperforms other methods.

Liu et al. [38] proposed multi-objective task scheduling based on particle swarm optimization (PSO) algorithm. This algorithm assigns the resources to the tasks to achieve an optimal makespan, resource utilization, and average consumed power. However, this algorithm fails to schedule the available tasks dynamically. Zhong et al. [39] suggested greedy particle swarm optimization to solve one of the main problems in a CC environment, namely, task scheduling. This algorithm shows good performance compared with existing techniques. Awad et al. [40] proposed task scheduling based on load balancing and particle swarm optimization, which includes consideration availability, reliability of the CC environment, and other traditional factors. The experimental results show that their proposed algorithm achieves better results than standard PSO.

Zuo et al. [41] proposed an adaptive multi-objective ant colony algorithm to schedule tasks in CC. This method is evaluated using four metrics: makespan, cost, deadline violation rate, and resource utilization. Moreover, this algorithm is superior to other similar techniques by approximately 56.6%.Meanwhile, a service flow scheduling model is suggested in [42] with different quality of service (QoS) requirements. Several properties of QoS in CC are taken into accounts, including security, reliability, and cost and response time. The simulation results illustrated that the suggested strategy is effective in terms of developing security data for dependent task scheduling in CC environment. However, the research in [43] is taken into consideration less requirements of QoS and the experimental results illustrated that the multiple pheromone algorithm (MPA) based on ant colony optimization (ACO) technique outperforms other existing algorithms in term of QoS and achieved optimal solutions.

Khalili and Babamir [44] introduced a novel of multi-objective dependent task scheduling based on parallel grey wolf optimization (PGWO) algorithm. The major targets of PGWO are minimizing both makespan and cost and maximizing throughput of resources. The experimental results showed that PGWO technique has better performance in terms of makespan, cost and throughput. Meanwhile, Natesan and Chokkalingam [45] focused on consumed energy and makespan to optimize CC task model. Based on simulation results, it is observed that the mean of the suggested GWO outperforms both standards GWO and PSO.

Manasrah and Ba Ali [46] proposed a new model based on the combination of GA and PSO algorithms. GA-PSO is considered as a multi-objective model because its targets are reducing makespan, cost and load balancing. The simulation outcomes showed that GA-PSO algorithm reduces the makespan of the dependent tasks. Moreover, it balances the whole load of system effectively. Finally, the simulation results illustrated that GA-PSO method and other compared techniques are convergent in terms of optimal solutions and high quality. However, Xue and Wu [47] introduced a novel design of QoS based on hybrid GA and PSO algorithms in order to schedule applications to available and suitable CC resources. In (GHPSO) technique, In GHPSO algorithm, mutation and crossover operations of GA are embedded into PSO algorithm. The simulation results showed that GHPSO technique provides superior results comparing with traditional PSO algorithm in terms of makespan and cost.

Yeboah and Odabi [48] employed ACO in hybrid bee ant colony (HBACO) algorithm in order to optimize the load balancing whereas ABC is utilized to improve task scheduling in CC environment. The experimental outcomes achieved better performance than Ant Colony algorithm and Bees Life algorithm in terms of load balancing, makespan, waiting time and response time. While Arunarani et al [49] suggested a combination of bat algorithm (BA) and firefly algorithm (FA) in order to optimize scheduling time, cost, deadline and risk rate.

Many studies apply metaheuristic optimization algorithms in order to solve task scheduling in CC environment. Thus, in this paper, a new metaheuristic optimization algorithm (VWOA) is designed and applied to solve this problem. VWOA includes mutation and other evolutionary operations in order to design the vocalization behavior of humpback whales. Thus, we apply VWOA to reduce the number of internal parameters; and investigate the performance of a basic version of VWOA for task scheduling in CC environment.

3. MULTI-OBJECTIVE MODEL FOR TASK SCHEDULING

Task scheduling is needed when many clients demand for the same resource simultaneously. The strategy of task scheduling relies on the fitness parameters. The fitness function depends on four attributes in this work, namely, resource utilization, energy, round trip time latency, and task weight. The fitness function returns at maximum value, as shown in function (3.1).

$$Fitness\ function\ (F) = \frac{1}{4}[\,FTw + Ru + (1 - E) + (1 - L)] \tag{3.1}$$

Where Ru refers to resource utilization, E represents the energy demanded to execute a task in a VM (Virtual Machine), L denotes the round-trip time latency, and FTw indicates total task weight based on two parameters: requester class and importance of execution task. The term $(1 - E)$ is added to the equation to highlight its extreme priority and latency.

In a CC environment, some tasks require many CPU resources, whereas other tasks demand less CPU resources and more storage. Ru contains two parameters: CPU cost (CCost) and memory cost (MCost); and they are expressed as Equations (3.2) and (3.3) [25, 41].

$$Ru = \frac{1}{2}[C_{cost}(j) + M_{cost}(j)] \quad (3.2)$$

$$C_{cost}(j) = C_{base} * C_j * t_{ij} + C_{Trans} \quad (3.3)$$

where Cbase represents the base cost when a resource is used by the lowest utilization to keep the machine running to receive incoming requests, tij indicates the duration time in which Ti (Task i) is processed in VMj (Virtual machine (j)), Cj indicates to the cost of VMj and CTrans is the cost associated with the CPU transmission to accomplish the required tasks. Equation (3.4) shows the memory cost [25, 41].

$$M_{cost}(j) = M_{base} * M_j * t_{ij} + M_{Trans} \quad (3.4)$$

Where the base cost of memory is denoted as Mbase, Mj indicates the cost of memory of VMj, tij represents the duration time of task (Ti) running in VMj, and MTrans is a constant value that represents the cost associated with the memory transmission.

A relationship exists between computing time and energy consumption [50]. Thus, the amount of energy that is required by Ti in VMj is defined as in [51] by Equation (3.5). The total energy consumed by VMj to finish all tasks is defined by [25] as in Equation (3.6).

$$E_{cost}(j) = ECT_{ij} * PO^{j}_{exe} \quad (3.5)$$

Where ECT_{ij} indicates energy consumption time, and PO^{j}_{exe} represents the power of VMj that is consumed during execution Ti.

$$\text{TE (j)} = \sum_{j=1}^{|VM|} E_{cost}(\text{j}) \quad (3.6)$$

Round trip time represents the latency (L) time of the whole step, including the start of the transmission, reception and waiting time for a response. Latency is calculated in [40] using Equation (3.7).

$$L = EET + ETT + 2 * delay \quad (3.7)$$

Where EET indicates expected execution time, whereas the expected transmission time is denoted as ETT.

EET depends on four main factors: CPU speed, RAM speed, task size, and bandwidth. When the CPU speed and RAM speed are high, the task is processed in a small amount of time. In other words, an inverse relationship is found between CPU and RAM speeds and processing time of task. The execution time of short or medium-sized tasks is less time consuming compared with that when the task size is long. EET and ETT are expressed in Equations (3.8) and (3.9), respectively [40].

$$EET = CPU_{Speed} + RAM_{Speed} + {}^{Task\ size}/_{Bandwidth} \quad (3.8)$$

$$ETT = {}^{Task\ size}/_{Bit\ rate} \quad (3.9)$$

Final task weight (FTw) is an important factor that is affected by prioritization of tasks. This attribute contains two main parameters: client type and task weight. Client types represent the privilege classes of clients. Moreover, it consists of number of classes (such as class A, class B, class C, and class D; as in Table 3.1). In case a client decides to join one of the previous classes, the client should participate as a member by paying a fee and using the services that are offered weekly or monthly depending on the size of tasks and the amount of resources that will be reserved. Each task has a priority weight that gives the task a high, medium, or low priority to use the available resources as in Table1.The priority weight factors that are used on the proposed mathematical model are 0.4, 0.3, 0.2, and 0.1.These values are chosen such that their summation is equal to one. The final task weight equation is expressed as in Equation (3.10).

$$FTw = CT * Tw \quad (3.10)$$

Where CT indicates the client type and Tw presents task weight.

Table 1: Weights for four tasks with their classes and schedule prioritization

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

4. VOCALIZATION OF HUMPBACK WHALE OPTIMIZATION

In this section, the inspiration of the vocalization behavior of humpback whales is presented. The mathematical model for VHWO is provided.

4.1.Humpback vocalization in nature

Humpback whales (Megaptera novaeangliae) are considered medium-sized baleen whales and one of the biggest mammals in the world [52]. This type of mammal does not completely sleep because it comes up to the surface of oceans to breathe. Thus, only half of its brain sleeps. Humpback whales are considered highly smart with high emotional intelligence.

Humpback whales are also known for their foraging behavior. They employ a special hunting and feeding method called the bubble-net technique [25, 53, 54]. These whales prefer to chase and entrap their prey that are close to the surface of the ocean. Looking for prey is achieved by making particular bubbles along a circle or spiral shape [25, 55].

Humpback whales are also known for their vocalization behavior [52]. Humpback whales produce a large variety of vocal sounds, which are loud and repetitive chains known as songs [52, 56]. The functions of humpback whale songs are for mating and migration. Only male humpback whales sing; females do not. While singing, singers are stable at the range of depth 8–30 m, with their heads oblique about 45° [52]. Moreover, these whales swim and dive slowly due to the self-noise disturbance exposure of echoes.

The singers are males who are responsible for decision making about mating and migration. Male humpback whales sing the same songs to expedite discovery of echoes from groups. Singers progressively alter the structure and organization of their songs throughout the winter season; modifications occur rapidly in several years than others [52, 56]. In addition, they sing for about 40 minutes, remain quiet for about 10 min, and repeat singing for about 40 min [52]. Singers stop singing when they pinpoint the position of another lone singer by hearing its song, deciding to join its cluster, and swimming at a high speed, which leads to its self-noise that hinders detection of echoes [52, 56]. Singing adults assist other lone adults to join the cluster they're associated with by singing with a singer who is the leader of the cluster.

Cow–calf represents a female accompanied by a male. The number of males exceeds the number of females by at least two years. Each female has at least two escorts: the primary one swims nearest to her, and the secondary escort stays at a farther distance. In case the secondary escort tries to unsettle the primary escort from his position, they attack each other. The winner will be near the female, and the loser will leave the cluster. In other words, females help by attracting escorts to join clusters. The rest of the cluster comprises non-singing adults, lone females, and escorts. Non-singing adults are associated with the cluster for migration. Although females never sing, singers detect them during singing and enjoin them to their cluster. Escorts almost never sing as well and help look for females (Fig. 1).

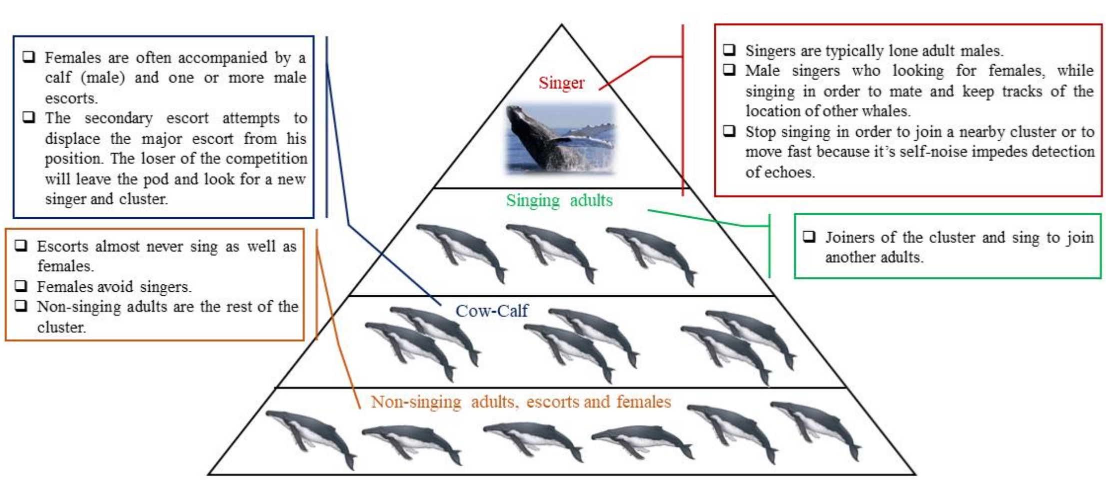

Fig. 1: Social hierarchy of humpback whales for mating and migration.

According to [51, 56], the main stages of humpback whale singing behavior are as follows.

1. Singing and looking for females.
2. Listening, tracking, and approaching the nearest singer.
3. Joining the cluster of the closest singer.
4. Competing of escorts.
5. Forming a cluster.

4.2. Mathematical model and VWOA

Humpback whales can pinpoint the location of any object by singing and echolocation. As shown in Fig.2, sound waves spread in the region while a male humpback whale (singer) sings. Thus, whales that hear the song will join the pod. Moreover, a singer can identify the location of a female humpback whale so that she may become part of his pod (Fig.3).

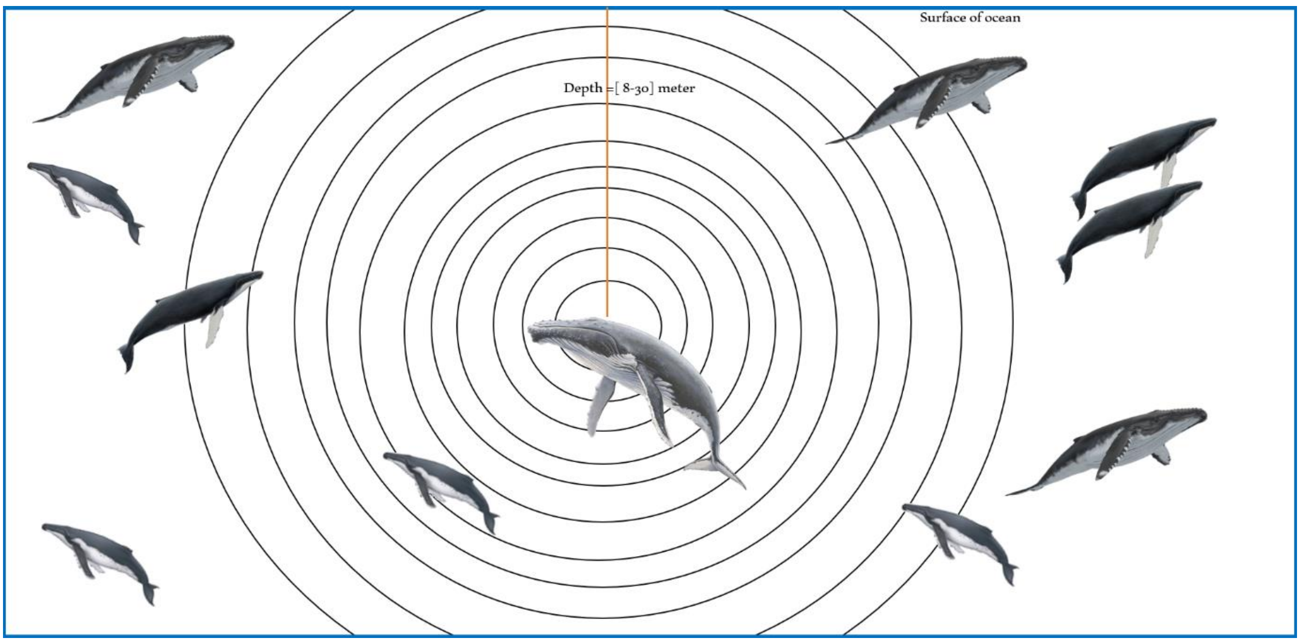

Fig. 2: Male humpback whale in the center, singing

The fittest solution is considered the singer (S_m), whereas the second and third best solutions are singing adults (S_a) and cow–calf (C_c), respectively. The remaining members of the pod represent the rest of the candidate solutions.

When an adult whale hears a song, he approaches the singer and joins his pod. Moreover, joiner whales update their position permanently depending on the position of the singer to be in the pod. In case an adult whale hears more than two songs, he will approach the nearest one (Fig. 4).In other words, a whale will calculate the distance between his position and location of singers that hears his vocalization using Equation (4.1).

$$Distance = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2 + (Z_2 - Z_1)^2} \tag{4.1}$$

Where (X_1, Y_1, Z_1) and (X_2, Y_2, Z_2) indicate the position coordinates of singer whales and the whale who wants to join the pod, respectively. As shown in Fig. 4, an adult whale indicates to a whale that decides to join the pod. Singers A and B represent singers that are considered leaders at their pods.

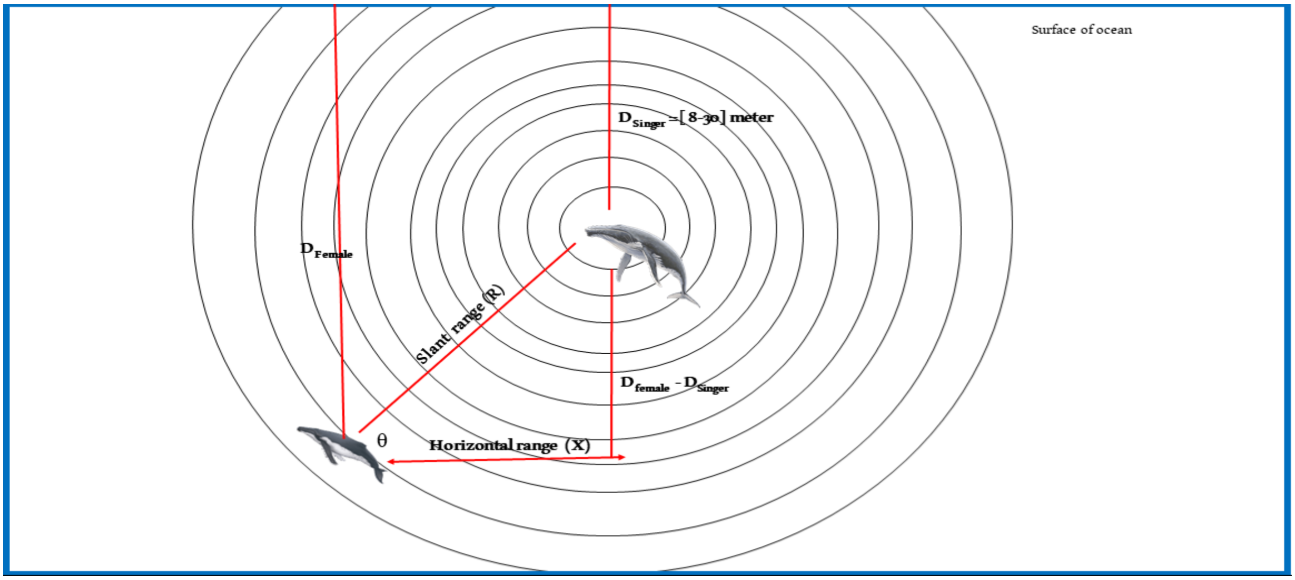

Fig. 3: A singer detects a female

The echolocation technique of male humpback whales is similar to the echolocation method of killer whales but with different functions [39, 52, 58]. Humpback whales use sounds for communicating, whereas killer whales use sounds for hunting. As shown in Fig. 3, the formulas in (4.2) are presented in this regard according to [58].

$$\theta = \sin^{-1}\left(\frac{|\vec{d}_{female}-\vec{d}_{Singer}|}{\vec{R}}\right) = \tan^{-1}\left(\frac{|\vec{d}_{female}-\vec{d}_{Singer}|}{\vec{X}}\right) \qquad (4.2)$$

Where d_{female} and d_{Singer} represent the depth of the female humpback whale and the singer humpback whales sonars, respectively, R is the slant range between a female humpback and singer humpback, and X is the horizontal range. The angle between slant and horizontal ranges is denoted by θ [58].

When a singer detects a female, he will approach her using Equations (4.3), (4.4), and (4.5).

$$\theta^* = Sin^{-1}\left(\frac{|\vec{d}_{female}-\vec{d}_{Singer}|}{\vec{A}(\vec{R}-\vec{a})}\right) \qquad (4.3)$$

Vectors $\vec{A}$ and $\vec{a}$ are calculated as in Equations (4.4) and (4.5), respectively.

$$\vec{A} = 2\vec{r} \qquad (4.4)$$

$$\vec{a} = \frac{1}{2}\vec{n} \qquad (4.5)$$

Where, each $\vec{r}$ is decreased linearly from 1to -1 over the course of iteration and $\vec{n}$ is a random vector between 0 and 1.

When a random value of $\vec{A}$ is in the interval [-2r, 2r], the position of a search agent is updated between its current position and the female's position. In other words, when $|\vec{A}|$ >1, the singer approaches the female. When $|\vec{A}|$ <1, the singer moves away from the female.

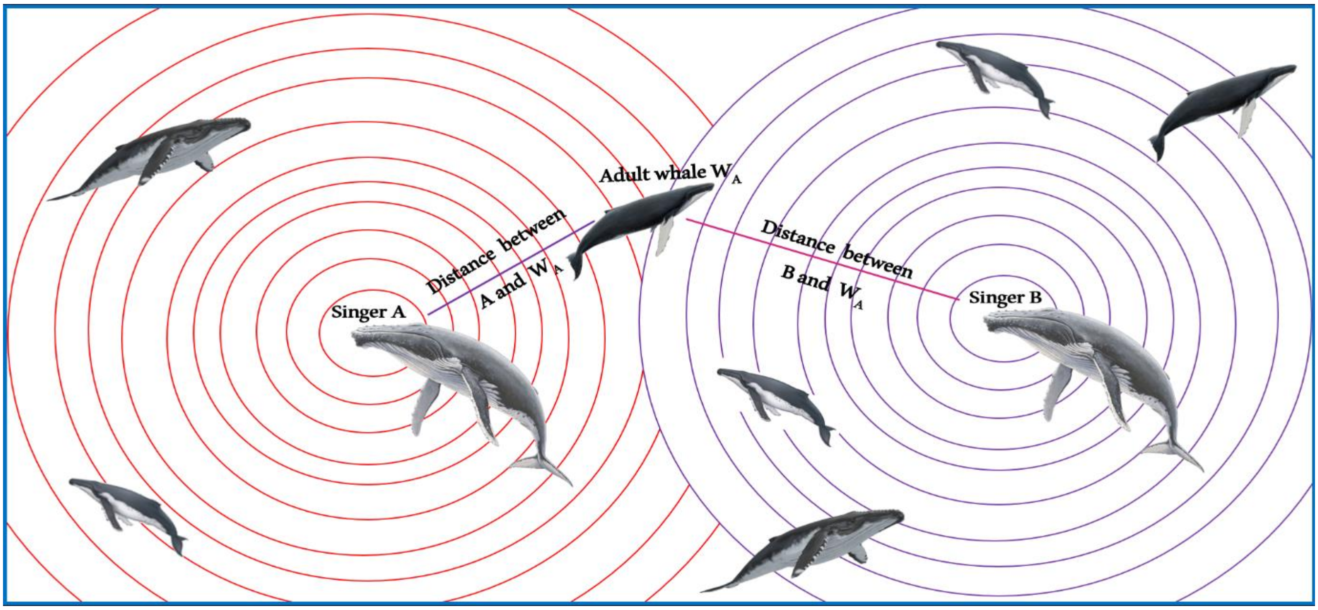

Fig. 4: Singing of two male humpback whales

When the singer approaches the detected female, the joiners will update their positions depending on the position of the singer to encircle the female and force her to join their pod. In mathematical simulating, this behavior $S_m$is assumed to be the best candidate solution. Let S_a and $C_c$be the second and third best solutions, respectively. Thus, the first three best solutions obtained are saved; and force the other whales including non-singing adults, lone females, and escorts to update their positions depending on the position of the best search agents. This behavior is modeled mathematically as in Equations (4.6), (4.7), and (4.8).

$$\theta_{S_m} = Sin^{-1}\left(\frac{|\vec{d}_F - \vec{d}_{S_m}|}{\vec{A}(\vec{R}_{F-S_m} - \vec{a})}\right) \quad (4.6)$$

$$\theta_{S_a} = Sin^{-1}\left(\frac{|\vec{d}_F - \vec{d}_{S_a}|}{\vec{A}(\vec{R}_{F-S_a} - \vec{a})}\right) \quad (4.7)$$

$$\theta_{C_c} = Sin^{-1}\left(\frac{|\vec{d}_F - \vec{d}_{C_c}|}{\vec{A}(\vec{R}_{F-C_c} - \vec{a})}\right) \quad (4.8)$$

Where d_{S_m}, d_{S_a}, and d_{C_c} indicate the depth of the singer, singing adults, and cow–calf, respectively. $R_{F\_S_m}, d_{S_a}$, and $R_{F\_C_c}$ represent the slant range between a female and the singer, singing adults, and cow–calf, respectively.

In order to mathematically model approaching the female, the parameter $(\vec{r})$ is decreased. Meanwhile, the variation range of $(\vec{A})$ is minimized by $(\vec{r})$. Fig.4.5 (a) illustrates that |A| > 1 obliges the whales to move towards the female and force her to join their pod.

Humpback whales generally search with respect to the location of the three best solutions (S_m, S_a and C_c). They splay from each other's to search for a female and converge to force the female to join their pod when it gives up and stops swimming. In order to mathematically design divergence, $\vec{A}$ is employed with random values greater than 1 or less than -1 to force the search agent to move forwards or to move away from the female. This confirms exploration and permits VWOA to search globally. Fig. 5 (b) illustrates that |A|<1 obliges the humpback whales to go away from the female hopefully to detect another female especially when other whales force the female to join their pod. Another significant parameter that supports exploration phase is $\vec{a}$. As seen in Equation (4.5), the vector $(\vec{a})$ includes random values in [0, 1]. This parameter supplies random weights for females to randomly emphasize or deemphasize the influence of females in determining the distance in Equation (4.3). This helps VWOA to display more random behavior during optimization, supporting exploration and avoid local optima. Moreover, ($\vec{a}$) is selected randomly from 0 to 1 at all times to confirm exploration including the initial iteration and final iterations. Thus, this parameter is useful in case of stagnation of local optimal, especially at the final iteration.

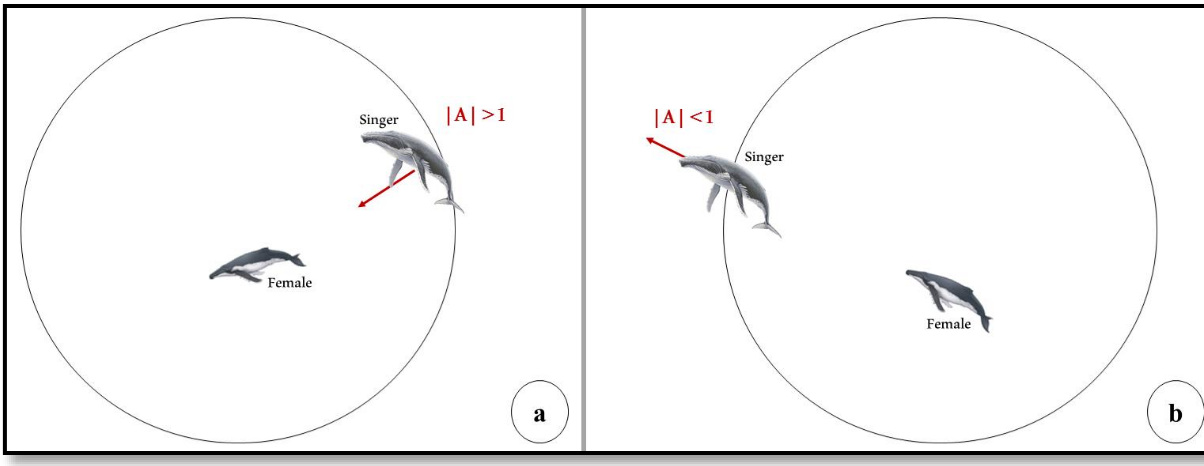

Fig. 5: (a) joining a pod versus (b) Searching for a female to join

To sum up, the exploration phase starts with generating a random population of humpback whales (candidate solutions) in the VWOA mechanism. Over the course of iterations, S_m, S_a and C_c whales deduce the potential location of the female. Each candidate solution updates its distance from the female. Candidate solutions tend to space from the female when |A| <1 and move towards the female when |A| > 1.

The implementation process of the VWOA is presented as a pseudocode in Fig. 6.

Input: *A, a* and *maximum number of iterations //* where *A* and *a* are coefficient vectors.
Output: Optimal solution (S_m)
Initialize the humpback whales population W_j (j= 0, 1, 2, ………..,n)
Initialize: A, a and I =0 // Where l is number of iteration.
Calculate the fitness function of each search agent using Equation (3.1)
S_m = the first best search agent
S_a= the second-best search agent
C_c= the third best search agent
Do {
  For each search agent
    Update the position of the current search agent by Equations. (4.6-4.8)
  End For
  Update A and a
  Calculate the fitness function of all search agents
  Update θ_{S_m}, θ_{S_a} and θ_{C_c}
  I= I+1
 }
While (I < maximum number of iteration)
}
Return S_m

Fig. 6: Abstraction of VWOA

The following points are presented to show the ability of the novel metaheuristic optimization algorithm (VWOA) to solve optimization problems:

o The social hierarchy helps VWOA to maintain the best solutions during the course of iterations.

- The suggested hunting technique uses the candidate solutions in order to find the position of the prey.
- To help VWOA to transit smoothly between exploration and exploitation; the parameters' values of A and n are utilized.
- Half of iterations for exploration phase and other half for exploitation phase.
- There are two main parameters of VWOA to be adjusted; r and c.

4.3. VWOA scheduler

The vocalization of the humpback WOA is considered a global optimizer because it involves the ability of both phases: exploration and exploitation. Moreover, it is used to solve a combinatorial optimization problem. The scheduling process is completed by simulating the vocalization behavior of humpback whales. First, the population of search agents is initialized. For each search agent, the fitness function is computed by using the proposed multi-objective task scheduling model. Second, the first, second, and third search agents (S_m, S_a, and C_c) are initialized. Other search agents update their positions depending on the position of the best search agent. In case |A| > 1, the search agent approaches the female; otherwise; the search agent goes away from the best search agent. The algorithm stops when the optimal solution is reached.
To schedule the submitted tasks through the VWOA, the phases are as follows. First, input the number of tasks, task deadline, type of class, priority, the number of VMs, their abilities, and other requested and relevant parameters. Second, each task represents a humpback whale. When a task T_i is successfully assigned to a virtual machine Vm_j, it will be recorded into the queue. This process is repeated for all subsequent tasks. For each task, the fitness value will be calculated. Considering the makespan and load balancing, the maximum value will be obtained.

5. WOA

WOA was used to allocate tasks to VMs [37]. WOA starts with a random population of solutions, whereas the current solution is considered the best solution; and continues the iteration operation based on the current solution. This operation is repeated until it reaches the best solution. The following steps are the main steps of WOA [59].

- *Initialization phase*: In this stage, the population of search agents W_i (i= 1, 2, 3..., k) is randomly created.
- *Fitness calculation phase*: Function (3.1) is used to calculate the fitness function. The best search agent W* is chosen based on evaluation.
- *Encircling prey phase:* In this phase, the position of a prey is assumed to be fixed. Thus, the prey is surrounded by humpback whales assuming that the current solution is the best solution (prey). Other whales (search agents) update their locations based on the current best agent, which is represented as in Equation (5.1).

$$\vec{P} = \left|\vec{C} . \overrightarrow{W^*}(t) - \vec{W}(t)\right| \tag{5.1}$$

Where the current iteration is denoted as t, $\overrightarrow{W^*}(t)$ indicates the position vector of the best solution, and $\vec{W}(t)$ represents the position vector of a search agent. Thus, the other search agents update their positions based on the best search agent using Equations (5.2), (5.3), and (5.4).The value of $\vec{r}$ decreases from 2 to 0.

$$\vec{W}(t+1) = \overrightarrow{W^*}(t) - \vec{R} . \vec{P} \tag{5.2}$$

Where $\vec{C}$ and $\vec{R}$ indicate the coefficient vectors.

$$\vec{R} = 2\vec{r} . \vec{n} - \vec{r} \tag{5.3}$$

$$\vec{C} = 2.\vec{n} \tag{5.4}$$

- *Exploitation phase*: In this stage, two basic techniques will be chosen. The first technique is called shrinking encircling. In this technique, the algorithm calculates the new position of a search agent. The value of a search agent $\vec{r}$ should be a value between [-1, 1]. The new value should be calculated depending on the initial location of the search agent and the current best agent. The second technique is spiral updating position. It updates the position of agents in the spiral method using Equation (5.5).

$$\vec{W}(t+1) = \left|\overrightarrow{W^*}(t) - \overline{W(t)}\right| . e^{bl} . \cos(2\pi l) + \overrightarrow{W^*}(t) \tag{5.5}$$

Where l is a value in [-1, 1], and b is a constant for determining the shape of the logarithmic spiral.

- *Exploration phase*: The position of the search agent will be updated according to a randomly selected search agent using Equations (5.6) and (5.7).

$$\vec{P} = \left|\vec{C} \cdot \overrightarrow{W_{rand}} - \vec{W}\right| \qquad (5.6)$$

Where $\vec{W}$ indicates the position vectors of other search agents except $\overrightarrow{W_{rand}}$ which represents the random position vector.

$$\vec{W}(t+1) = \overrightarrow{W_{rand}} - \vec{R} \cdot \vec{P} \qquad (5.7)$$

- *Termination phase*: In case a search agent exits in the search region, the value of the best search agent will be updated and the next iteration begins. This process will stop when the best solution is found.

Fig.7 presents the pseudocode for the suggested WOA scheduler for task scheduling in a cloud computing environment. The main objective of the suggested scheduler is to assign tasks to the available and suitable VM optimally. WOA scheduler is based on the proposed multi-objective task scheduling model and WOA.

```
Input: Tasks (T1, T2, T3, ……..,Tn), Virtual machines (V1, V2, ……. , Vk)
Output: Allocated tasks to the virtual machines
BEGIN
Initialize the population
Calculate the fitness of each search agent using Equation (5.1)
Initialize the current best agent W*
While (I< maximum number of iteration)
  For each search agent
   Update P, R, C, l and r.
    If (P <0.5)
         If (|R|<1)
              Update the position of the current search agent using Equation (5.2)
         Else if (|R|>1)
              Update the position of the search agent according to the randomly selected agent using Equation
               (5.7).
        End if
    End if
    If (P >= 0.5)
             Update the position of search agent using Equation (5.5)
    End if
  End for
  If (Any search agent goes beyond the search space and amends it)
     Update W*
     I= I+1
 End while
Return W*
```

Fig.7: Abstraction of WOA scheduler

.

6. EXPIREMENTAL RESULTS

The performance of WOA and the suggested VWOA for independent task scheduling is experimentally evaluated. Experimentation is conducted using a personal computer with Intel Core i-7 processor, 16 GB RAM, and Windows 10 operating system. The experiments employ CloudSim tool with JAVA. The results of the VWOA are compared with the findings of the existing WOA and RR techniques in terms of makespan, degree of imbalance, cost, resource utilization, and energy consumption.

A. Evaluation Metrics

1. *Makespan* indicates the total execution time that is required to execute all independent tasks [60, 61].
2. *Cost* represents the execution cost of execution tasks on specific VMs. In addition, cost depends on length of tasks, cost of data transfer, and storage [61, 62].
3. In *Resource utilization*, increasing the utilization of resources is beneficial to cloud computing service providers [62, 64]. These resources are wholly utilized to bring maximum profits by renting limited resources to clients.
4. In *Energy consumption*, the utilization of both CPU and resources affects power consumption by a task. Consumed energy is high if a CPU is not employed. However, consumed power is sometimes high because of the heavy request of resources [63, 65, 66].
5. *Degree of Imbalance (DI)* measures the imbalance among VMs [67] using Equation (6.1).

$$Degree\ of\ Imbalance\ (DI) = \frac{T_{max} - T_{min}}{T_{average}} \quad (6.1)$$

Where T_{max} indicates the maximum execution time of VMs, T_{min} represents the minimum execution time of VMs, and $T_{average}$ denotes the average execution time of VMs.

B. Results

This study compares the results of the VWOA with the results of the traditional WOA and RR scheduling algorithms over various independent tasks (100–500) with different number of VMs (8, 16, and 32), which were selected in previous studies [10, 20-22, 24]. The setup includes various numbers of tasks (100, 200, 300, 400, and 500), two data centers, two hosts, and 32VMs. Each scenario is implemented 30 runs. The average of each scenario is computed.
VWOA and WOA are metaheuristic optimization algorithms. VWOA mimics the vocalization behavior of humpback whale; while WOA mimics the hunting behavior of humpback whale. Thus, by using Equation (6.2) [68], the improvement in percentage of VWOA over WOA for 32 VMs is calculated. The percentage improvement for 100, 200, 300, 400 and 500 cloud tasks are 7.5%, 2.53%, 3.6%, 2.41%, 2.59% respectively, due to the high exploration and exploitation of VWOA. The exploration phase indicates to the procedure of seeking the promising regions of the search space as broadly as possible. However, exploitation phase indicates to the capability of local search around the promising regions that acquired in the exploration stage. VWOA has high exploration and exploitation ability due to the assistants who increase the search spaces and oblige females to join their groups. Moreover, it has stochastic operators in order to search globally and randomly in the search spaces.

$$Improvement = \left(1 - \frac{\sum Makespan_{VWOA}}{\sum Makespan_{WOA}}\right) * 100 \quad (6.2)$$

The simulation results are illustrated in Figs.8 - Fig.10. These figures show that the VWOA has better performance than traditional WOA and RR in terms of makespan due to the exploration and exploitation ability of VWOA. Exploration phase relies on the number of humpback whales in each cluster which is unlimited. It is contrast to some metaheuristic optimization algorithms such as GWO; where the number of search agents in each group between 5 and 12. At the initial phases of iterations, the equation (4.2) demands humpback whale to proceed randomly around each other's. While, the equation (4.6-4.8) permits other whales to relocation themselves or move in circle shaped toward the best search agent which leads whale to have high exploitation. Thus, the parameter |A| helps VWOA to provide good exploration, exploitation and local optimal avoidance. While RR algorithm is the worst performance because RR scheduling algorithm depends on time quantum' size and does not consider the information of tasks and resources. In case the time quantum is large, this leads for high waiting time. However, if the quantum time is small, this leads for low waiting time and too many tasks switches. This leads for high makespan.

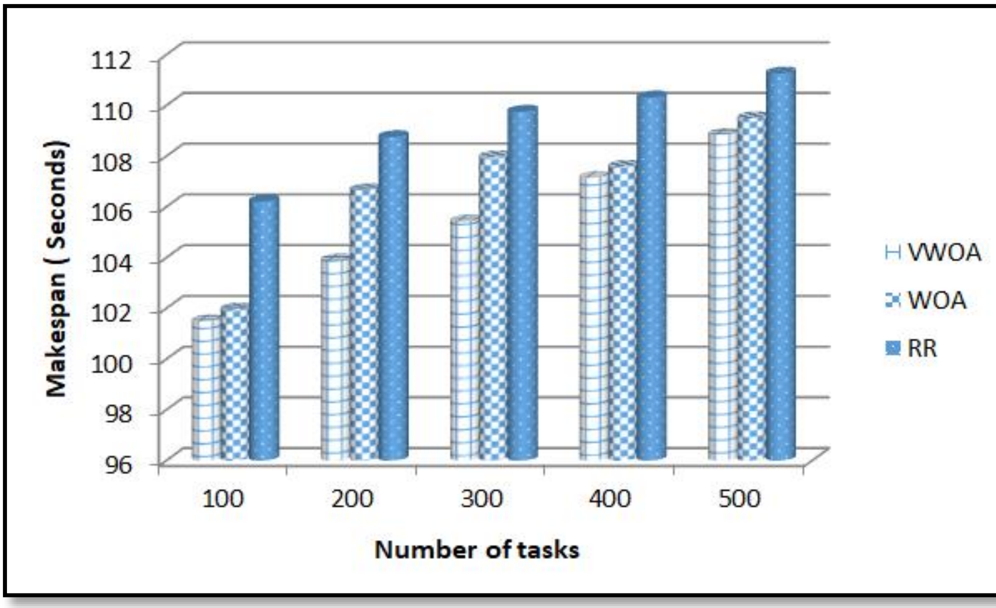

Fig.8: Comparing Makespan versus Number of independent tasks when number of VMs = 8.

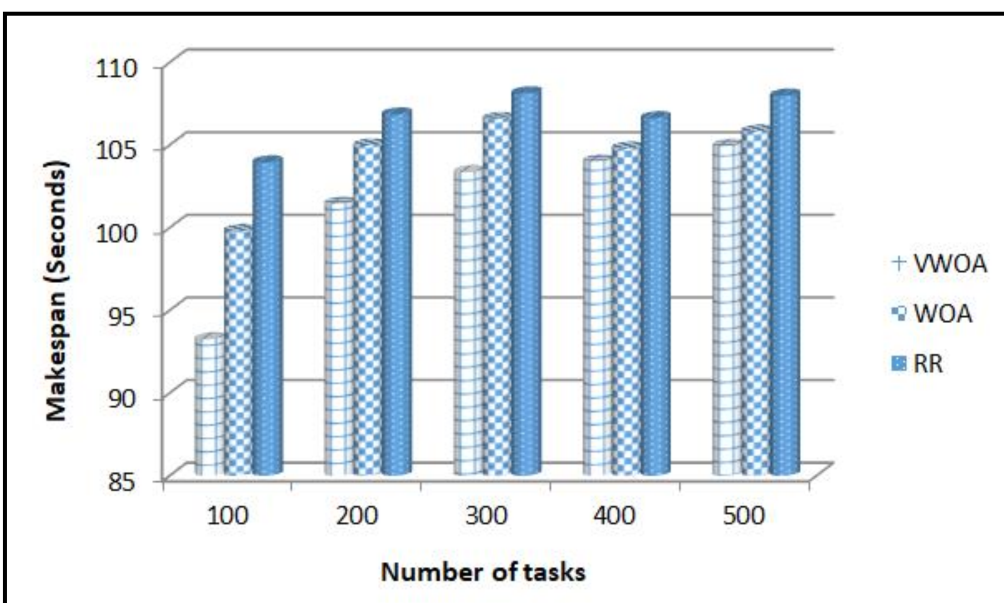

Fig.9: Makespan versus Number of independent tasks when VMs = 16.

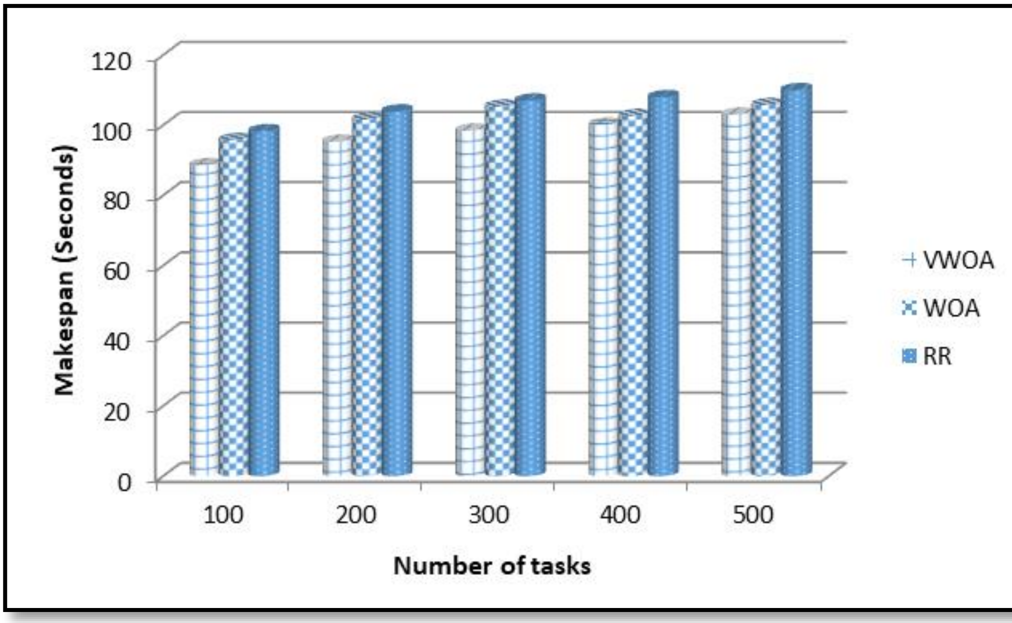

Fig.10: Makespan versus Number of independent tasks when number of VMs = 32.

Fig.11 shows that the experiments results of degree of imbalance when the three algorithms were tested for number of tasks (100, 200, 300, 400, and 500) on 32 VMs. It is obvious that VWOA is more efficient as well as has less degree of imbalance compared to WOA and RR algorithms. The mechanism designates tasks that have high weights to the available and powerful VMs with considering multi factors such as length of tasks, importance of execution task and ability of VM. Therefore, the overall execution time of each VM will be minimized. DI depends on makespan as seen in Equation (6.1). This is why each algorithm has its DI.

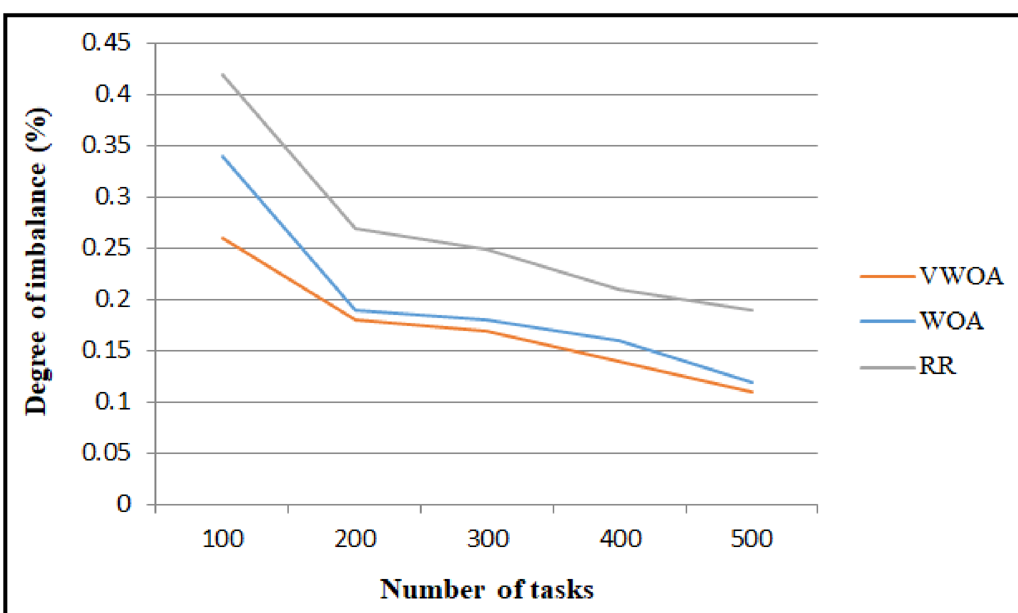

Fig. 11: Average DI of VWOA, WOA, and RR when the tasks were tested on 32 VMs

Fig.12 shows the average energy consumption of 32 VMs. VWOA consumes less energy than the existing WOA and RR algorithms. In particular, VWOA saves 17% and 72% energy compared with WOA and RR algorithm, respectively. Energy consumption depends on the overall execution time. Thus, VWOA consumed less energy due to its low makespan. However, RR algorithm consumed more energy due to its high makespan.

Figs. 13-15 illustrate the scheduling cost of various numbers of independent tasks on different numbers of VMs (8, 16, and 32). The number of independent tasks plays a significant role in performance and number of available resources. In other words, when the number of tasks is very large, reaching a global optimal solution becomes complex. By contrast, when the number of available resources is very large, a global optimal solution can be easily obtained because a large number of tasks and limited available resources lead to increase in makespan, scheduling cost, and waiting time. However, as shown in Figs.6.6-6.8, VWOA demonstrates better performance than the existing techniques. The total execution cost of scheduling the tasks of VWOA is reduced by 13% and 22% compared with those of WOA and RR algorithm, respectively.

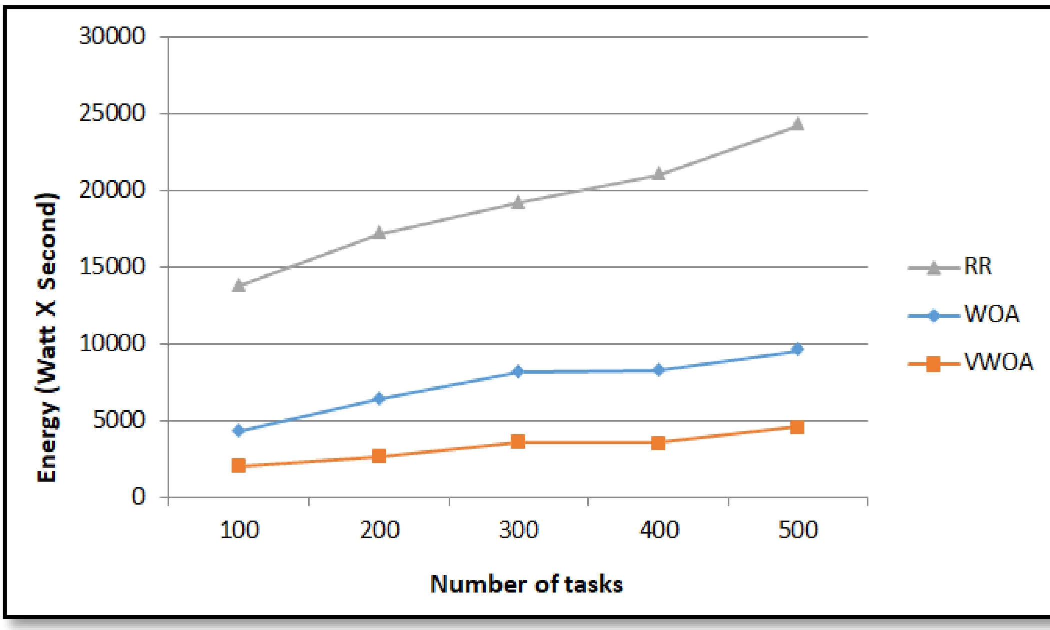

Fig. 12: Energy results for various numbers of tasks on 32 VMs

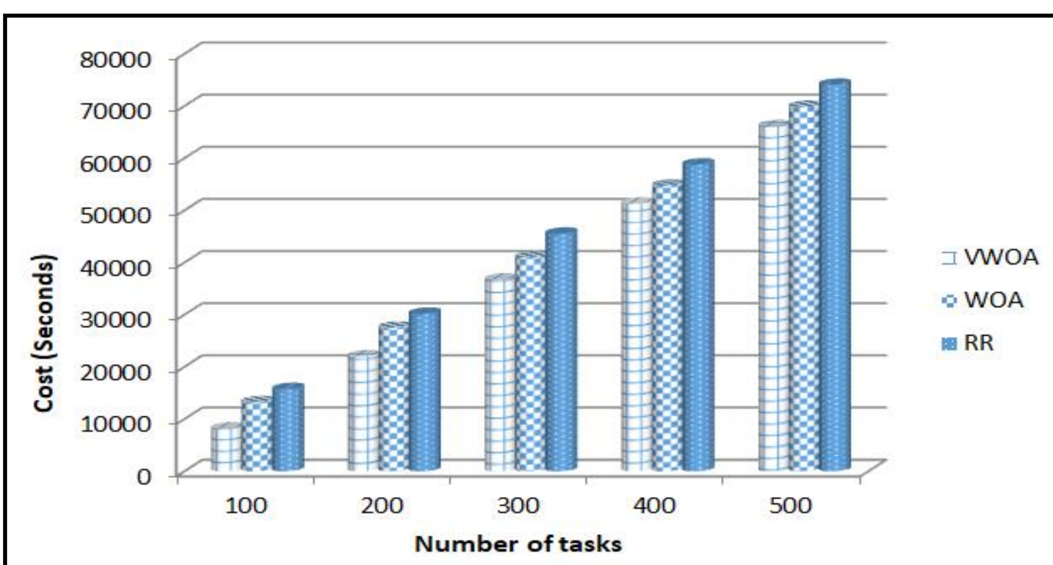

Fig.13: Cost versus number of tasks on 8 VMs.

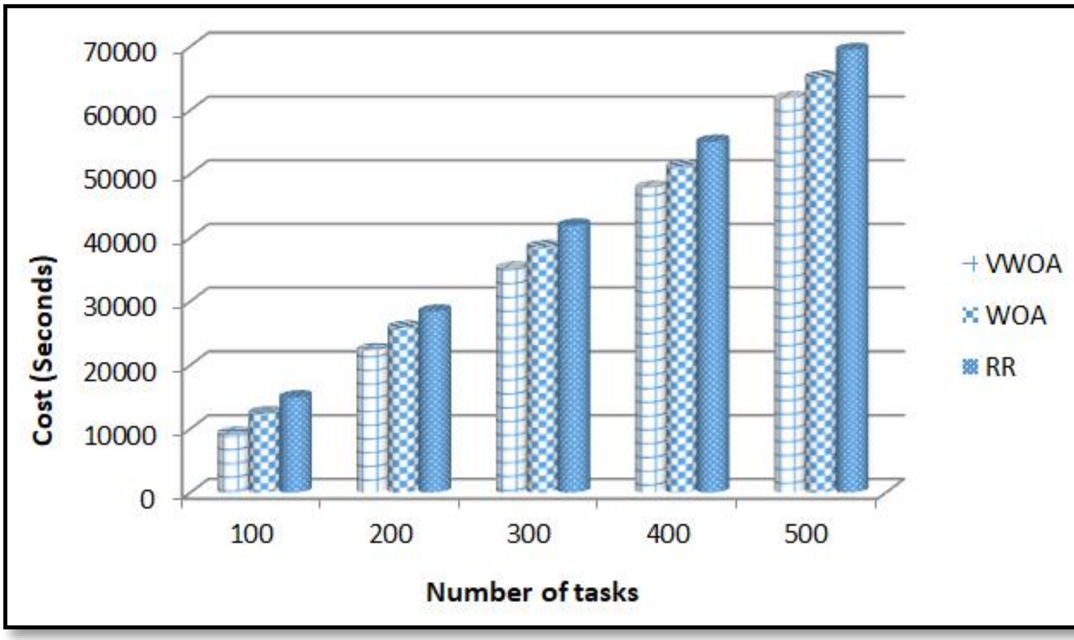

Fig.14: Cost versus number of tasks on 16 VMs.

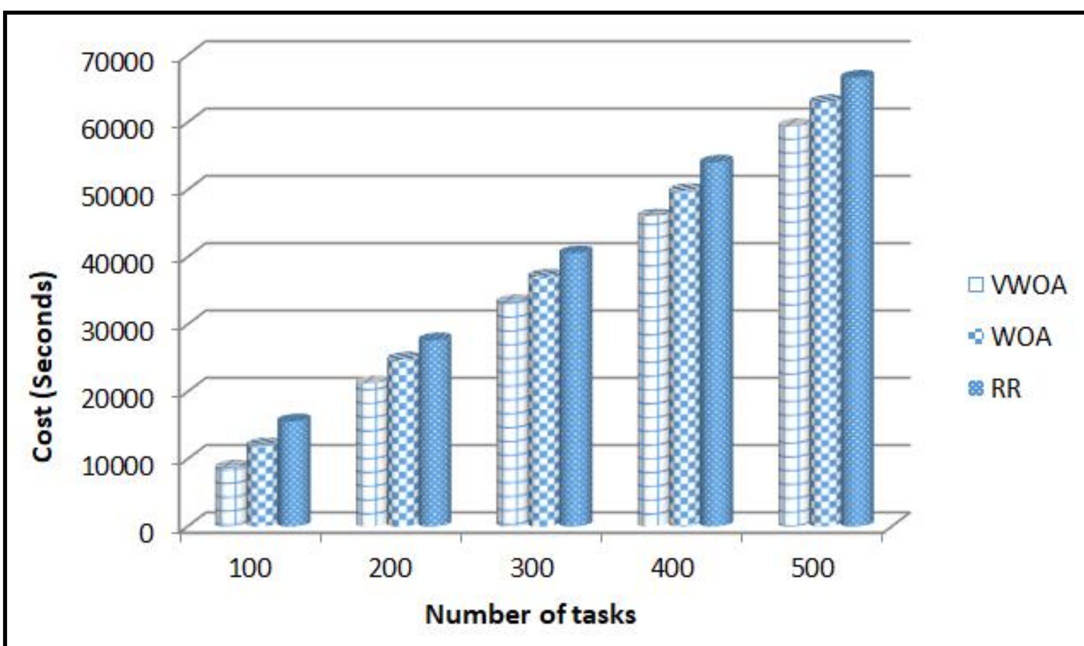

Fig.15: Cost versus number of tasks on 32 VMs.

Figs. 16 and 17 represent the resource utilization of the VWOA, original WOA, and RR algorithms using 16 and 32 VMs and various numbers of tasks, respectively. VWOA clearly utilizes more resources than the original WOA and RR algorithm. The average resource utilization of VWOA is increased by 8% and 35% compared with those of WOA and RR algorithms, respectively. VWOA utilized more resources due to its high exploration and exploitation. In addition, it has updated information about all tasks and resources; which contrast to RR algorithm that has not any information about resources and tasks. RR algorithm assigns tasks to VMs as a circle, does not consider assigning tasks to suitable VM.

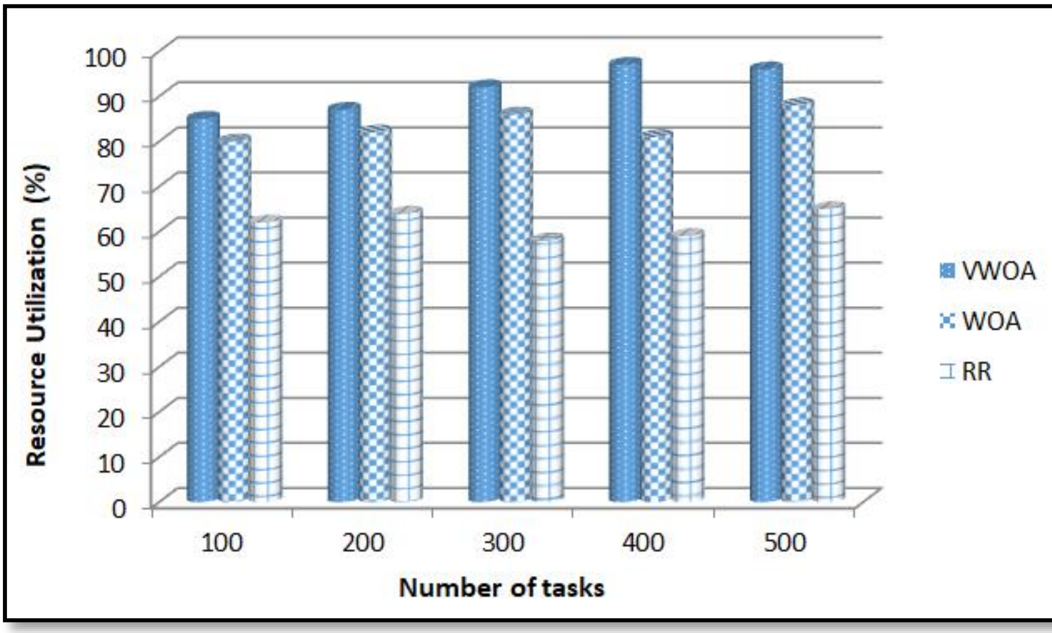

Fig. 16: Comparison of the utilization of number of tasks on 16 VMs.

.

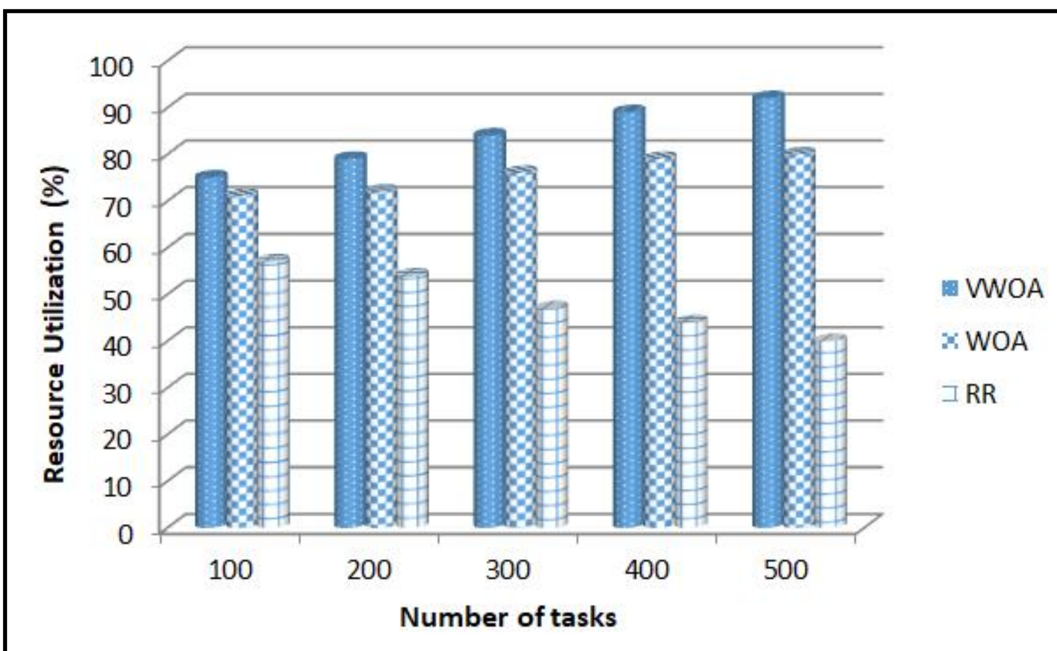

Fig.17: Comparison of the utilization when VWOA, WOA, and RR were tested on different number of tasks on 32 VMs.

7. Conclusion

Many heuristic and metaheuristic algorithms are adopted to improve cloud computing task scheduling. In this paper, a new metaheuristic algorithm VWOA is designed and implemented, which mimics the vocalization behavior of humpback whales. This algorithm supposes that there are three best solutions at each cluster. In other words, there are three best local optimal solutions. Thus, the first present solution is considered the best solution, and the global optimal solution is found based on the search agent. The multi-objective mathematical model is adopted to improve independent task scheduling in a CC environment. The major objectives of VWOA are to minimize cost, makespan, and energy consumption and maximize resource utilization. CloudSim tool was used to evaluate the algorithm. Simulation results clearly show that the presented VWOA algorithm provides superior results compared with the existing WOA and RR algorithms in terms of makespan, cost, degree of imbalance, energy consumption, and resource utilization. Moreover, the three best solutions found at each cluster can produce additional solutions with optimal scheduling.

In future, our work will be extended for dependent tasks. Moreover, the proposed VWOA can be adapted to solve other optimization problems.

REFERENCES

[1]: Mell, P., &Grance, T. (2009). The NIST definition of cloud computing. National institute of standards and technology, 53(6), 50.
[2]:Mell, P., &Grance, T. (2011). The NIST definition of cloud computing.USNat'l Inst. of Science and Technology; http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.
[3]: Zanoon, N., Al-Haj, A., &Khwaldeh, S. M. (2017). Cloud Computing and Big Data is there a Relation between the Two: A Study. International Journal of Applied Engineering Research, 12(17), 6970-6982.
[4]:Tarhini, A., Al-Badi, A., Almajali, M., & Alrabayaah, S. (2017). Factors influencing employees' intention to use cloud computing. Journal of Management and Strategy, 8(2), 47.
[5]:Lin, X., Wang, Y., Xie, Q., &Pedram, M. (2015). Task scheduling with dynamic voltage and frequency scaling for energy minimization in the mobile cloud computing environment. IEEE Transactions on Services Computing, 8(2), 175-186.
[6]: Sharieh, A., & Albdour, L. (2017, December). "A Heuristic Approach for Service Allocation in Cloud Computing," International Journal of Cloud Applications and Computing, 7 (4), 60-74.
[7]:Navimipour, N. J. (2015). A formal approach for the specification and verification of a trustworthy human resource discovery mechanism in the expert cloud. Expert Systems with Applications, 42(15-16), 6112-6131.
[8]:Malawski, M., Juve, G., Deelman, E., &Nabrzyski, J. (2015). Algorithms for cost-and deadline-constrained provisioning for scientific workflow ensembles in IaaS clouds. Future Generation Computer Systems, 48, 1-18.
[9]:Navimipour, N. J., Rahmani, A. M., Navin, A. H., &Hosseinzadeh, M. (2015). Expert Cloud: A Cloud-based framework to share the knowledge and skills of human resources. Computers in Human Behavior, 46, 57-74.
[10]:Rimal, B. P., Jukan, A., Katsaros, D., &Goeleven, Y. (2011). Architectural requirements for cloud computing systems: an enterprise cloud approach. Journal of Grid Computing, 9(1), 3-26.
[11]:Rimal, B. P., Choi, E., &Lumb, I. (2009, August). A taxonomy and survey of cloud computing systems.In INC, IMS and IDC, 2009.NCM'09. Fifth International Joint Conference on (pp. 44-51). IEEE.
[12]: Abu Khurma,R., AL Harahsheh, H., & Sharieh, A. (2018, October)."Task Scheduling Algorithms in Cloud Computing: Case Study on Modified Round Robin," Journal of Theoretical and Application Information Technology, 96 (17), 5869-5888.
[13]:Navin, A. H., Navimipour, N. J., Rahmani, A. M., &Hosseinzadeh, M. (2014). Expert grid: new type of grid to manage the human resources and study the effectiveness of its task scheduler. Arabian Journal for Science and Engineering, 39(8), 6175-6188.
[14]:Atiewi, S., Yussof, S., &Ezanee, M. (2015). A comparative analysis of task scheduling algorithms of virtual machines in cloud environment. Journal of Computer Science, 11(6), 804 -812.
[15]: B. Mahafzah, B., & Jaradat, B.(2010). "The hybrid dynamic parallel scheduling algorithm for load balancing on chained-cubic tree interconnection networks" Journal of Supercomputing, 52(3), 224-252.
[16]: Mahafzah, B., & Jaradat, B.(2018). "The load balancing problem in OTIS-Hypercube interconnection networks" Journal of Supercomputing, 46(3), 276-297.
[17]:Ullman, J. D. (1975). NP-complete scheduling problems. Journal of Computer and System sciences, 10(3), 384-393.
[18]:Xu, Y., Li, K., He, L., & Truong, T. K. (2013). A DAG scheduling scheme on heterogeneous computing systems using double molecular structure-based chemical reaction optimization. Journal of Parallel and Distributed Computing, 73(9), 1306-1322.
[19]:Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in Engineering Software, 2016 (95), 51-67.
[20]: Masadeh, R., Alzaqebah, A&Hudaib, A. (2018). Grey Wolf Algorithm for Requirements Prioritization. Modern Applied Science, 12(2), 54.
[21]: Barham, R., Sharieh, A., &Sliet, A. (2016). Chemical reaction optimization for max flow problem. IJACSA) International Journal of Advanced Computer Science and Applications, 7(8). 189-196
[22]: Alshraideh, M., Mahafzah, B., Eyal Salman, H., & Salah, I. (2013). "Using genetic algorithm as test data generator for stored PL/SQL program units" Journal of Software Engineering and Applications, 6(2), 65-73.
[23]: Khanafseh, M., Surakhi, O., Sharieh, A., & Suliet, A. (2017). "A Comparison between Chemical Reaction Optimization and Genetic Algorithms for Maximum Flow Problem," International Journal of Advance Computer Science and Applications (IJACSA), 8 (8), 7-15.
[24]:Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. Information Sciences, 2014 (270), 255-287.
[25]:Sreenu, K., & Sreelatha, M. (2017). "W-Scheduler: whale optimization for task scheduling in cloud computing". Springer, Cluster Computing, 1,8- 19.
[26]: Alshraideh, M., Mahafzah, B., &Al-Sharaeh, S. (2011). "A multiple-population genetic algorithm for branch coverage test data generation" Software Quality Journal, 19(3), 489-513.
[27]:Barham, R., Sharieh, A., & Sleit, A. (2018). "A meta-heuristic framework based on clustering and preprocessing dataset for Solving

the link prediction problem," Journal of Information Science, 2018, 1-24.
[28]: Maqableh, M., Karajeh, H., & Masadeh, R. (2014). Job scheduling for cloud computing using neural networks. Communications and Network, 6(3), 191-200.
[29]:Khan, M. A. (2012). Scheduling for heterogeneous systems using constrained critical paths. Parallel Computing, 38(4-5), 175-193.
[30]:Gupta, S., Kumar, V., & Agarwal, G. (2010, February). Task scheduling in multiprocessor system using genetic algorithm. In Machine Learning and Computing (ICMLC), 2010 Second International Conference on (pp. 267-271). IEEE.
[31]: Masadeh, R., Sharieh, A., &Sliet, A. (2017). Grey wolf optimization applied to the maximum flow problem. International Journal of Advanced and Applied Sciences, 4(7), 95-100.
[32]: Keshanchi, B., Souri, A., &Navimipour, N. J. (2017). An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing. Journal of Systems and Software, 2017 (124), 1-21.
[33]:He, H., Xu, G., Pang, S., & Zhao, Z. (2016). AMTS: Adaptive multi-objective task scheduling strategy in cloud computing. China Communications, 13(4), 162-171.
[34]:Xu, X., Cao, L., & Wang, X. (2016). Resource pre-allocation algorithms for low-energy task scheduling of cloud computing. Journal of Systems Engineering and Electronics, 27(2), 457-469.
[35]:Yuan, H., Bi, J., Tan, W., & Li, B. H. (2017). Temporal Task Scheduling With Constrained Service Delay for Profit Maximization in Hybrid Clouds. IEEE Trans. Automation Science and Engineering, 14(1), 337-348.
[36]: Kaur, S., &Verma, A. (2012).An efficient approach to genetic algorithm for task scheduling in cloud computing environment. International Journal of Information Technology and Computer Science (IJITCS), 4(10), 74-79.
[37]:Reddy, G. Narendrababu, and S. Phani Kumar. "Multi objective task scheduling algorithm for cloud computing using whale optimization technique."International Conference on Next Generation Computing Technologies. Springer, Singapore, (2017), 286-297.
[38]: Liu, J., Luo, X. G., Zhang, X. M., & Zhang, F. (2013). Job scheduling algorithm for cloud computing based on particle swarm optimization.In Advanced Materials Research 2013(662), 957-960. Trans Tech Publications.
[39]: Zhong, Z., Chen, K., Zhai, X., & Zhou, S. (2016). Virtual machine-based task scheduling algorithm in a cloud computing environment. Tsinghua Science and Technology, 21(6), 660-667.
[40]:Awad, A. I., El-Hefnawy, N. A., &Abdel_kader, H. M. (2015). Enhanced particle swarm optimization for task scheduling in cloud computing environments. Procedia Computer Science, 2015 (65), 920-929.
[41]:Zuo, L., Shu, L., Dong, S., Zhu, C., & Hara, T. (2015). A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. IEEE Access, 2015 (3), 2687-2699.
[42]:Liu, W., Peng, S., Du, W., Wang, W., & Zeng, G. S. (2014). Security-aware intermediate data placement strategy in scientific cloud workflows. Knowledge and information systems, 41(2), 423-447.
[43]: Gogulan, R., Kavitha, A., & Kumar, U. K. (2012). An multiple pheromone algorithm for cloud scheduling with various QOS requirements. International Journal of Computer Science Issues (IJCSI), 9(3), 232-238.
[44]:Khalili, A., &Babamir, S. M. (2017). Optimal scheduling workflows in cloud computing environment using Pareto-based Grey Wolf Optimizer. Concurrency and Computation: Practice and Experience, 29(11), 1 -11.
[45]:Natesan, G., &Chokkalingam, A. (2018). Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. ICT Express.
[46]:Manasrah, A. M., & Ba Ali, H. (2018). Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. Hindawi, Wireless Communications and Mobile Computing, 2018, 1-16.
[47]:Xue, S. J., & Wu, W. (2012). Scheduling workflow in cloud computing based on hybrid particle swarm algorithm. Indonesian Journal of Electrical Engineering and Computer Science, 10(7), 1560-1566.
[48]:Yeboah, T., &Odabi, O. I. (2015). Hybrid bee ant colony algorithm for effective load balancing and job scheduling in cloud computing. West African Journal of Industrial and Academic Research, 13(1), 54-59.
[49]: Arunarani, A. R., Manjula, D., &Sugumaran, V. (2017). FFBAT: A security and cost-aware workflow scheduling approach combining firefly and bat algorithms. Concurrency and Computation: Practice and Experience, 29(24), 1-13.
[50]:Bunde, D. P. (2006, July). Power-aware scheduling for makespan and flow.In Proceedings of the eighteenth annual ACM symposium on Parallelism in algorithms and architectures (pp. 190-196).ACM.
[51]: Sharma, M., & Garg, R. (2017, December). Energy-aware whale-optmized task scheduler in cloud computing. In 2017 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 121-126).IEEE.
[52]: Helweg, David A., et al. "Humpback whale song: Our current understanding." Marine mammal sensory systems.Springer, Boston, MA, 1992. 459-483.
[53]: Masadeh, R., Alzaqebah, A., & Sharieh, A. (2018). Whale Optimization Algorithm for Solving the Maximum Flow Problem. Journal of Theoretical & Applied Information Technology, 96(8), 2208- 2220.
[54]: Alzaqebah, A., Masadeh, R., &Hudaib, A. (2018, April). Whale Optimization Algorithm for Requirements Prioritization. In Information and Communication Systems (ICICS), 2018 9th International Conference on (pp. 84-89). IEEE.
[55]: Barham, R., Sharieh, A., & Sleit, A. (2018). "Moth Flame Optimization Based on Golden Section Search and its Application for Link prediction Problem," Modern Applied Science, 13(1): 10-27.
[56]: Maybaum, Hilary L. "Responses of humpback whales to sonar sounds." The Journal of the Acoustical Society of America 94.3 (1993): 1848-1849.
[57]: Lakshmi, R. D., &Srinivasu, N. (2016). A dynamic approach to task scheduling in cloud computing using genetic algorithm. Journal of Theoretical & Applied Information Technology, 85(2), 124- 135.
[58]: Totok R. Biyantoa,*, Matradjia , Sonny Irawanc , Henokh Y. Febriantod , NaindarAfdannya , Ahmad H. Rahmana , Kevin S. Gunawana , Januar A. D. Pratamaa , Titania N. Bethianab.(2017). Killer Whale Optimization Algorithm.Elsevier.(2017),1-7.
[59]: Masadeh, R., Hudaib, A., & Alzaqebah, A. (2018). WGW: A Hybrid Approach Based on Whale and Grey Wolf Optimization Algorithms for Requirements Prioritization. Advances in Systems Science and Applications, 18(2), 63-83.

[60]:Wu, Q., Yun, D., Lin, X., Gu, Y., Lin, W., & Liu, Y. (2012, May). On workflow scheduling for end-to-end performance optimization in distributed network environments.In Workshop on Job Scheduling Strategies for Parallel Processing. Springer, Berlin, Heidelberg, 76-95.

[61]: Atiewi, S., Yussof, S., Ezanee, M., &Almiani, M. (2016, April). A review energy-efficient task scheduling algorithms in cloud computing. In Systems, Applications and Technology Conference (LISAT), 2016 IEEE Long Island , 1-6.

[62]:Singh, P., Dutta, M., & Aggarwal, N. (2017). A review of task scheduling based on meta-heuristics approach in cloud computing. Knowledge and Information Systems, 52(1), 1-51.

[63]:Wang, X., Wang, Y., & Zhu, H. (2012). Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm. Mathematical Problems in Engineering,2012, 1-16.

[64]: Al-Shaikh, A., Khatab, H., Sharieh, A., & Sleit, A. (2016). "Resource Utilization in Cloud Computing as an Optimization Problem", International Journal of Advanced Computer Science and Applications, 7 (6), 336-342.

[65]:Atiewi, S., Abuhussein, A., & Saleh, M. A. (2018, September). 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.

[66]:Atiewi, S., Yussof, S., Rusli, M. E. B., &Zalloum, M. (2018). A power saver scheduling algorithm using DVFS and DNS techniques in cloud computing data centres. International Journal of Grid and Utility Computing, 9(4), 385-395.

[67]: Almezeini, N., & Hafez, A. (2017). Task scheduling in cloud computing using lion optimization algorithm. International Journal of Advanced Computer Science and Applications, 8 (11), 77-83.

[68]:Yin, H., Wu, H., & Zhou, J. (2007, August). An improved genetic algorithm with limited iteration for grid scheduling.In Grid and Cooperative Computing, 2007.GCC 2007. Sixth International Conference on (pp. 221-227). IEEE.