

Improving Quality of Service in Cloud Computing Frameworks Using Whale Optimization Algorithm



Maher Ali Ibrahim^{1,2*}, Inas Anouar Al-Tahar³, Hasan Mohamed Salamah², Naji Ibrahim Mohamad⁴

¹ Computer Technology Engineering Department, College of Technical Engineering, Al-Ayen Iraqi University, Thi-Qar 64001, Iraq

² Information Technology Engineering Department, College of Information and Communication Technology Engineering, Tartous University, Tartous 96343, Syria

³ Biomedical Engineering Department, College of Engineering, University of Thi-Qar, Nasiriyah 64001, Iraq

⁴ Communication Technology Engineering Department, College of Information and Communication Technology Engineering, Tartous University, Tartous 96343, Syria

Corresponding Author Email: maheralia1978@gmail.com

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/isi.290526>

ABSTRACT

Received: 15 March 2024

Revised: 19 July 2024

Accepted: 28 September 2024

Available online: 24 October 2024

Keywords:

Quality of Service, cloud computing, task scheduling, virtual machines, metaheuristic algorithms, cloud platforms

Quality of Service is one of the most important research topics in cloud computing, both from the customer's point of view and the cloud service provider's point of view, due to the increasing number of cloud services and applications, along with the significant increase in users and workloads. Task scheduling has been a topic of discussion in many researches, some of which have proposed new ways to improve the Quality of Service in cloud systems. Recently, metaheuristic algorithms have been employed to improve job execution efficiency in such systems, which has proven effective in finding optimal task scheduling solutions. During this research, the latest metaheuristic algorithm Whale Optimization Algorithm (WOA) will be applied for optimizing task scheduling in cloud systems. Additionally, a multi-objective model for optimization will be used to achieve a balance between user satisfaction with these systems, and the requirements needed by service providers. Research results indicate WOA superiority in minimizing the execution time cost, price cost, and total cost parameters as contrasted with the current metaheuristic algorithms, which ultimately result in higher service quality.

1. INTRODUCTION

With the significant growth in apps considering the notable increase in application locations, data, and internet access, cloud computing has become increasingly popular and widespread. Cloud computing offers many advantages such as flexibility, scalability, and simplified service delivery to users, who no longer must have the underlying technologies to use cloud computing services, through a pay-per-use model and release these resources when finished [1].

Recently, there has been a focus on how to allocate and distribute user tasks to the resource pool in a cloud system, known as task scheduling methodology. The goal is to improve performance, increase profits, and return on investment for cloud service providers and users alike [2].

Task scheduling is a complex problem, with its complexity degree being NP-complete, making it impossible to solve in linear time. Scheduling algorithms will fail as the problem size and dimensions increase. Therefore, improvements require finding an effective task scheduling strategy that increases the number of tasks while lowering the total system cost [3].

Because they can find the best answers to difficult optimization problems, metaheuristic algorithms have attracted a lot of attention lately. These algorithms, which

include the bee algorithm, the ant algorithm [4], the bat algorithm, the genetic algorithm, and other algorithms, are being used to tackle challenging optimization issues through collaboration and individual competition [5].

One of the latest metaheuristic algorithms is the WOA which is influenced by the hunting and eating habits of humpback whales. Adopting and applying this method could lead to significant benefits in this field of research, then it could be a good choice for scenarios requiring in-depth exploration of the study area and potential solutions. The study emphasizes the usage of a cloud computing system that uses a multi-objective optimization methodology to reduce expenses and speed up job execution.

The growing quantity and kinds of applications and services provided by cloud systems, the increased number of clients, and the activities requiring scheduling, have all resulted in the appearance of a significant problem. In particular, previous scheduling techniques are no longer sufficient, highlighting the need for more sophisticated and effective strategies to be developed for cloud computing work scheduling [6].

Using the m-file editor, the programming lines and algorithms unique to this study were created and put into practice within the MATLAB environment. The m-file is used for simulation modeling as well as algorithm creation,

analysis, and optimization. It makes it possible to execute these programs, carry out the simulation, clearly display pertinent graphical representations, and compare the effectiveness of various approaches. It becomes a vital research tool as a result.

The increased requirements for quality in the cloud, which includes serving customers as soon as possible, saving operational costs, and optimizing resource utilization, all require developing a better task scheduling mechanism. This research aims to explore the potential applications and employments of the WOA in the above described domain, to enhance its service quality.

The rest sections of this research are organized as follows: The second section presents the literature review. The third section presents the resources and methodology. The fourth section describes the proposed algorithm. The fifth section outlines the results and discussion. Finally, the sixth section presents the conclusions.

2. LITERATURE REVIEW

Elmougy et al. [7] proposed the SRDQ scheduling methodology, which combines the Round Robin (RR) algorithm and the Shortest Job First (SJF) algorithm. The study's results outperformed those of the SJF, RR, and TsPBR algorithms, but it partially caused the problem for long-term tasks.

In 2017, Zhang and Zhou conducted a study [8] aimed at improving task scheduling performance and reducing failure rates by classifying tasks using Bayesian methods based on historical scheduling data.

In 2019, Alworafi et al. [9] focused on the impact of resource efficiency on scheduling outcomes, aiming to maximize resource utilization. Their empirical results highlighted the effectiveness of the HSLJF algorithm in reducing makespan and response time while enhancing resource usage and productivity, outperforming traditional algorithms such as SJF and LJF.

Prasanna Kumar and Kousalya's 2020 study [10] demonstrated significant improvements in makespan values, specifically in the time required for the final cloudlet to complete. In 2021, Shafahi and Yari [11] introduced a novel algorithm based on the Ant Colony Optimization (ACO) technique to enhance the Quality of Service (QoS) metrics for both cloud service providers and users. This algorithm, when compared with Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and traditional ACO algorithms, showed superior performance in makespan time, execution time, load balancing, and resource utilization.

In 2022, Kumar et al. [12] developed a method for assigning jobs to virtual machines, which showed that their proposed approach surpassed existing algorithms in terms of overall execution time.

The 2023 study by Anbarkhan and Rakrouki [13] aimed to optimize execution time and cost, demonstrating that an improved PSO algorithm yielded better results in workflow task execution and convergence speed.

Task scheduling strategies remain a significant challenge in enhancing service quality within cloud systems due to the dynamic nature of cloud environments, their inherent constraints, and diverse job implementation requirements. Recent studies illustrate the effectiveness of metaheuristic algorithms in addressing these challenges. In light of these findings, the latest metaheuristic, the Whale Optimization

Algorithm, will be employed to further enhance service quality in cloud systems.

3. RESOURCES AND METHODOLOGY

3.1 Concept of task scheduling in cloud computing

The number of requests made by users is rising along with the number of people using cloud computing. This indicates that there are a lot more things that need to be planned. As a result, effective algorithms that satisfy the necessary Quality of Service requirements are crucial for cloud service providers as well as clients [14].

Task scheduling has become one of main study areas in cloud systems. It is regarded as one of the primary difficulties that cloud computing systems encounter [15]. The process of allocating a job that the cloud computing system receives to the best resource for its completion is known as task scheduling. The scheduling algorithm plays a crucial role in selecting the right Virtual Machine (VM) to carry out a certain activity.

An ideal algorithm provides optimal solutions for both, the service provider of the cloud, and the user requesting the service, satisfying client satisfaction on one hand and enhancing the cloud system's efficiency on the other. There are several proposed algorithms to solve scheduling problems, which can be classified into three groups as follows:

Traditional algorithms: Such as RR, LJF, and SJF [16].

Algorithms with Heuristics: As in Min-Min and Max-Min [17].

Metaheuristic algorithms: Recently, Metaheuristic algorithms have been applied, which perform better than traditional algorithms. Examples include the Bee Algorithm, The algorithms for ant colonies, particle swarms (PSOs), and genetics [18].

3.2 A multi-objective scheduling paradigm for cloud systems

In cloud-based computing systems, task scheduling requirements must address both system and customer perspectives. Therefore, this research employs a multi-objective model for task scheduling [19]. This model integrates parameters related to Quality of Service—such as execution time, load, and economic cost—into a unified objective function [20, 21]. This integration is crucial for preparing the function to serve as a fitness function within the WOA.

The suitability of this model for the algorithm was confirmed through experimental comparisons with multiple models. The primary stages of operation according to this model are depicted in Figure 1.

For a set of virtual machines that are accessible and a set of jobs that are arriving to the cloud system, tasks are assigned to resources using a specific method (initially assigned randomly) called the initial scheduling phase. The output of this phase can be represented by the matrix A_{nm} .

In the second phase, the final optimization function is determined, which consists of partial functions (The function of time cost f_1 , the function for load costing f_2 , and the price cost function f_3). Finally, the final optimization function is utilized as a fitness function in the WOA to obtain the optimized final scheduling process.

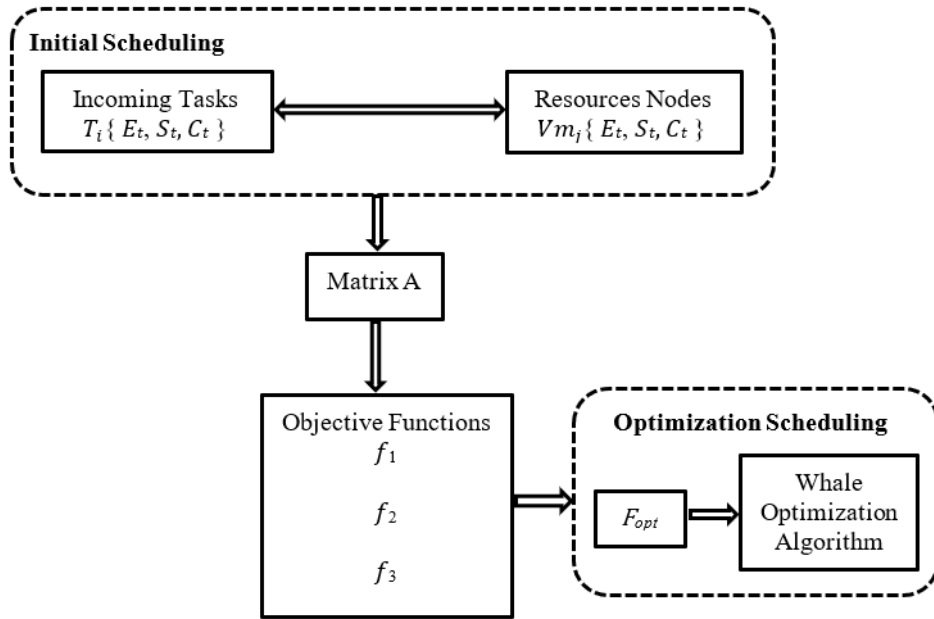


Figure 1. A logical view of the basic stages of work

For this multi-objective model, the system is expected to receive a total of tasks from users, which would include n . and each incoming task will be represented using the symbol T_i where i is the incoming task number to the system, resulting in $\{T_1, T_2, T_3, \dots, T_n\}$ set of tasks.

For this multi-objective model also, the cloud computing system consists of many computing nodes (Virtual Machines), The entire quantity of these nodes is supposed to be m , and each computing node will be represented using the symbol N_j where j is the node number in the system, resulting in $\{N_1, N_2, N_3, \dots, N_m\}$ set of nodes.

Tasks number is usually bigger than computing nodes number $n > m$. Thus, the resulting scheduling process is expressed by the matrix A_{nm} as follows:

$$A_{nm} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix}$$

a_{ij} is a decision variable and it takes value $a_{ij} = 1$ when j -th VM node is where j -th job is executed, if not $a_{ij} = 0$.

Processing capability, load capability, and resource bandwidth are three properties that may be used to indicate a cloud system's overall processing capacity and resource usage. The three vectors can be used to model the fundamental computer system: resource bandwidth vector C_n , load capability vector S_n , and processing capability vector E_n . Likewise, three vectors are employed in jobs: E_t , S_t , and C_t [22].

To handle a set of tasks that arrive at the system, it must process these chores with exceptional performance resource efficiency utilization, and cost reduction [19].

The function of time cost f_1 , the function for load costing f_2 , and the resource cost function represented by measuring power consumption and economic cost, are represented [22]. Therefore, the price cost will be represented by the function f_3 , based on the Eq. (1), Eq. (2), and Eq. (3) that follow:

$$f_1 = \sum_{i=1}^n \sum_{j=1}^m a_{ij} E_{t,i} / E_{n,j} \quad (1)$$

$$f_2 = \sum_{i=1}^n \sum_{j=1}^m a_{ij} S_{t,i} / S_{n,j} \quad (2)$$

$$f_3 = \sum_{i=1}^n \sum_{j=1}^m a_{ij} \frac{E_{t,i}}{E_{n,j}} \times \frac{C_{t,i}}{C_{n,j}} \times P \quad (3)$$

where, $E_{t,i}$ represents value of E_t for task i , $E_{n,j}$ represents the value of E_n for virtual machine j . Similarly, the parameters S and C are expressed in the same way.

Since the values of E_n , S_n , and C_n as well as E_t , S_t and C_t , are scaled differently. for each node and each task, this difference will introduce deviation in the paper for the optimal solution, this deviation will affect the final optimization process.

Therefore, normalization will be applied to obtain the values of the three functions indicated in Eq. (4), Eq. (5), and Eq. (6) respectively:

$$f_1 = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^m a_{ij} \frac{E_{t,i} / E_{n,j}}{\max_{v,i,j} \{E_{t,i} / E_{n,j}\}} \quad (4)$$

$$f_2 = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^m a_{ij} \frac{S_{t,i} / S_{n,j}}{\max_{v,i,j} \{S_{t,i} / S_{n,j}\}} \quad (5)$$

$$f_3 = \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^m a_{ij} \frac{(PE_{t,i} C_{t,j}) / (E_{n,j} C_{n,j})}{\max_{v,i,j} \{(PE_{t,i} C_{t,j}) / (E_{n,j} C_{n,j})\}} \quad (6)$$

The goal is to find a method for scheduling tasks that achieves best reduction in the values of the three aforementioned functions while taking into account the variations in cloud system requirements.

Each cloud system may have task execution requirements

where one system prioritizes minimizing time cost ($f1$) more than resource or price cost ($f3$), or vice versa.

Therefore, weights (w_i) will be added to the functions to control the degree of improvement for each function according to the cloud system's requirements. This will ultimately achieve the goal of the final optimization process.

The final optimization objective function is:

$$F_{opt} = \min\{w_1f_1 + w_2f_2 + w_3f_3\} \quad (7)$$

In this general case study, equal weights will be assigned to all functions ($w_1 = w_2 = w_3 = 1/3$). The final objective of the optimization process. Eq. (8) represents the job scheduling problem in the cloud.

$$F_{opt} = \min\left\{\frac{1}{3}(f_1 + f_2 + f_3)\right\} \quad (8)$$

4. PROPOSED ALGORITHM

WOA was motivated by humpback whale hunting techniques. To optimize numerical problems, Mirjalili and Lewis devised this approach in 2016 [23].

It has been tested on various complex optimization problems and has shown high performance, especially with issues involving multimodal functions.

Therefore, this algorithm will be dedicated to solving the job scheduling issue in the cloud. Humpback whales are among the most famous and largest predatory whales. They feed on small fish using a unique and intelligent feeding method called bubble-net feeding.

Humpback whales search for prey and then release spiral or path-shaped water bubbles to reduce the distance between them and their prey. They then attack and prey upon them. Figure 2 illustrates the feeding strategy of humpback whales [24].

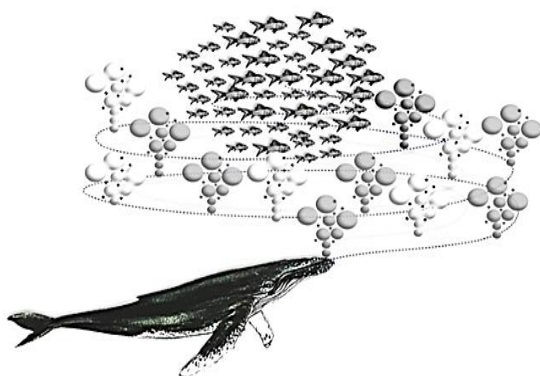


Figure 2. Feeding strategy of humpback whales

The humpback whale, acting as a search agent, offers a potential remedy for the optimization issue, the algorithm employs multiple search agents to determine the optimal solution, starting with a set of arbitrary fixes, the specific improvement rules change the proposed solution, until the termination clause is satisfied. Three primary steps are used by the humpback whale algorithm [23, 24]:

Encircling preying: Initially, the whales are not aware of the ideal place for the prey. When the target is encircled, the whales gradually approach and update their positions, the behavior of whales is represented by Eq. (9) and Eq. (10)

respectively:

$$\vec{D} = |C \times \vec{X}^*(t) - \vec{X}(t)| \quad (9)$$

$$\vec{X}(t + 1) = \vec{X}^*(t) - A \times \vec{D} \quad (10)$$

\vec{D} shows the distance vector between the search agent and the intended prey, \vec{X}^* is the optimal solution, \vec{X} is the vector of position, t denotes the current iterations number, C and A are the coefficient vectors, and the definition of their computations is:

$$C = 2 \times r \quad (11)$$

$$A = 2a \times r - a \quad (12)$$

The symbol a displays linearly dependent value based on t beyond the maximum iterations number t_{max} , value that was reduced from 2 down to 0, where r is a randomly selected number between 0 and 1, as demonstrated below:

$$a = 2 - \frac{2t}{t_{max}} \quad (13)$$

The Bubble-net Attack (stage of utilization): Whales exhibit two behaviours in which they surround their intended prey with copious bubbles: Encircling constriction behavior and spiral constriction behavior. The encircling constriction behavior is modeled based on the equations of prey encirclement in the initial stage, represented by Eq. (9) and Eq. (10). It can be observed that it is applied when $|A| < 1$. Consequently, the circle surrounding the prey gradually shrinks according to this behavior. On the other hand, the spiral constriction behavior calculates the distance between the current optimal whale and every other whale, then the whales move in spiral path, such behavior is modeled according to Eq. (14):

$$\vec{X}(t + 1) = \vec{D}' \times e^{lb} \times \cos(2\pi l) + \vec{X}^*(t) \quad (14)$$

$\vec{D}' = |X^*(t) - \vec{X}(t)|$ denotes the difference between the current whale and the best one, l denotes random number in the interval $[-1,1]$, b is a constant. To replicate the two actions in practical applications of the algorithm, a probability value P is assigned, which takes a random value between 0 and 1 [25]. Presumably, the likelihood of the whale upgrading its position according to the encircling behavior as well as the spiral contraction behavior is 0.5.

Search for Prey (Exploration Stage): Whales adopt this behavior in an attempt to expand the search range for prey and obtain a larger food quantity. This ensures the attainment of a globally optimal solution rather than a local one within the search space of the algorithm (when $|A| \geq 1$). According to this behavior, the best search agent available right now location is swapped out for a randomly chosen search agent location based on Eq. (15):

$$\vec{X}(t + 1) = \vec{X}_{rand} - A \times |C \times \vec{X}_{rand} - \vec{X}(t)| \quad (15)$$

where, \vec{X}_{rand} is the randomly chosen search agent's location vector.

Considering the previously, Figure 3 provides an illustration of the Whale Optimization Algorithm's primary phases.

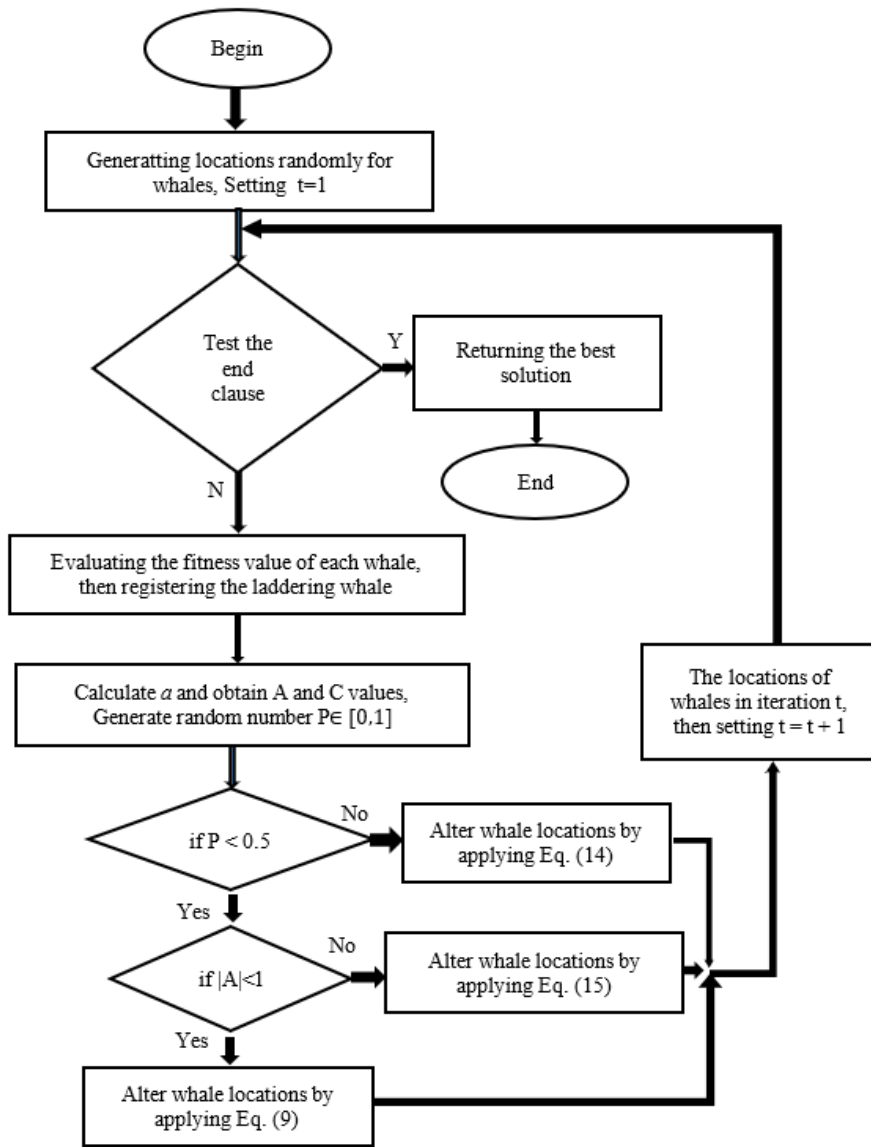


Figure 3. Whale Optimization Algorithm flow chart

4.1 Task and job scheduling based on WOA

Issue of task scheduling here is represented by set of tasks incoming to the cloud system, where each task has its execution requirements. Additionally, a limited number of virtual machines were used, each virtual machine has specifications such as CPU, memory size, and resources.

Table 1. Whale Optimization Algorithm parameters mapping into a task scheduling problem

Predation in Whales	Task Scheduling
Individuals (Whales)	Cloud tasks
Food search operations	Optimal solution search operations
Whale position	A_{nm} solution for the function F_{opt}
Leader whale position	Current optimal solution A_{nm} for F_{opt}
Fitness of whale	Value of function F_{opt}

A more efficient way is needed to assign incoming tasks as requests to available virtual machines in the cloud to achieve lowest total cost, which is expressed as a multi-objective function for optimization in cloud computing. For this research, a multi-objective optimization function will be employed as a fitness function, then the whale optimization

technique will be applied, mappings are listed in Table 1.

The flowchart in Figure 4 shows the essential working steps for applying the WOA in cloud computing, aimed at achieving the final optimal solution for the scheduling process. In iteration t of the WOA, each whale's position data may be transmitted after they have all updated their places to the A_{nm} a resolution for the assigned responsibilities.

Based on the principles in the A_{nm} matrix, we can determine the fitness value F_{opt} for every whale and identify the leading whale, which will have the smallest fitness function value. The location information of leader whale is used to update where required the locations of the other whales in the algorithm's next iteration. The previous stages and operations are continued till the very end of the iteration. The leading whale's last location data is transmitted to the A_{nm} matrix, and the cloud computing system will utilize this answer to develop an optimal job execution plan.

The steps for implementation are as follows:

Step 1: Initialization Phase: The purpose of this step is to start the execution process. It involves assigning tasks to cloud workers and whales, initializing the locations of the whales, and setting the fundamental execution parameters (search space dimension, maximum iterations number, and whales number).

Step 2: Search for the optimal scheduling solution: After the assignment between cloud tasks and whales is done, the process of searching for the optimal solutions according to WOA begins. The whale position information is transferred to the A_{nm} matrix, and the fitness function value for every whale is computed according to F_{opt} function. Then, the leader whale (which achieves the lowest fitness value) is selected, representing the best available solution as of now. Depending on the location information of the leader whale, global solutions are explored in the search space and local solutions are exploited according to the current value of a and the random value P , as follows:

If $P < 0.5$, the whale positions are updated according to the spiral constriction behavior Eq. (14). In the event not, A 's value is checked. If $|A| < 1$, whale positions are updated through the shrinking encircling mechanism described in Eq. (10). However, if $|A| \geq 1$, the whale locations are altered based on the location of a randomly selected whale (search

agent location), according to Eq. (15), to ensure the search for the best solution, which is global and optimal.

Step 3: Following an update to every whale's position, one iteration is completed. Once iterations maximum number is reached, the process of searching for the optimal scheduling solution will be ended. Next, the leading whale's location data is added to the decision factors aij , which represent the last resort for cloud computing's job scheduling procedure.

WOA application in the field of job scheduling in cloud systems offers numerous advantages that other metaheuristic algorithms cannot provide, particularly its high effectiveness in navigating the search space in pursuit of the best possible answer. This is due to its utilization of exploitation and exploration techniques, which gives it the capability and speed to reach optimal solutions. Additionally, its ability to work with increasing dimensions of the search space allows it to remain stable when solving complex problems.

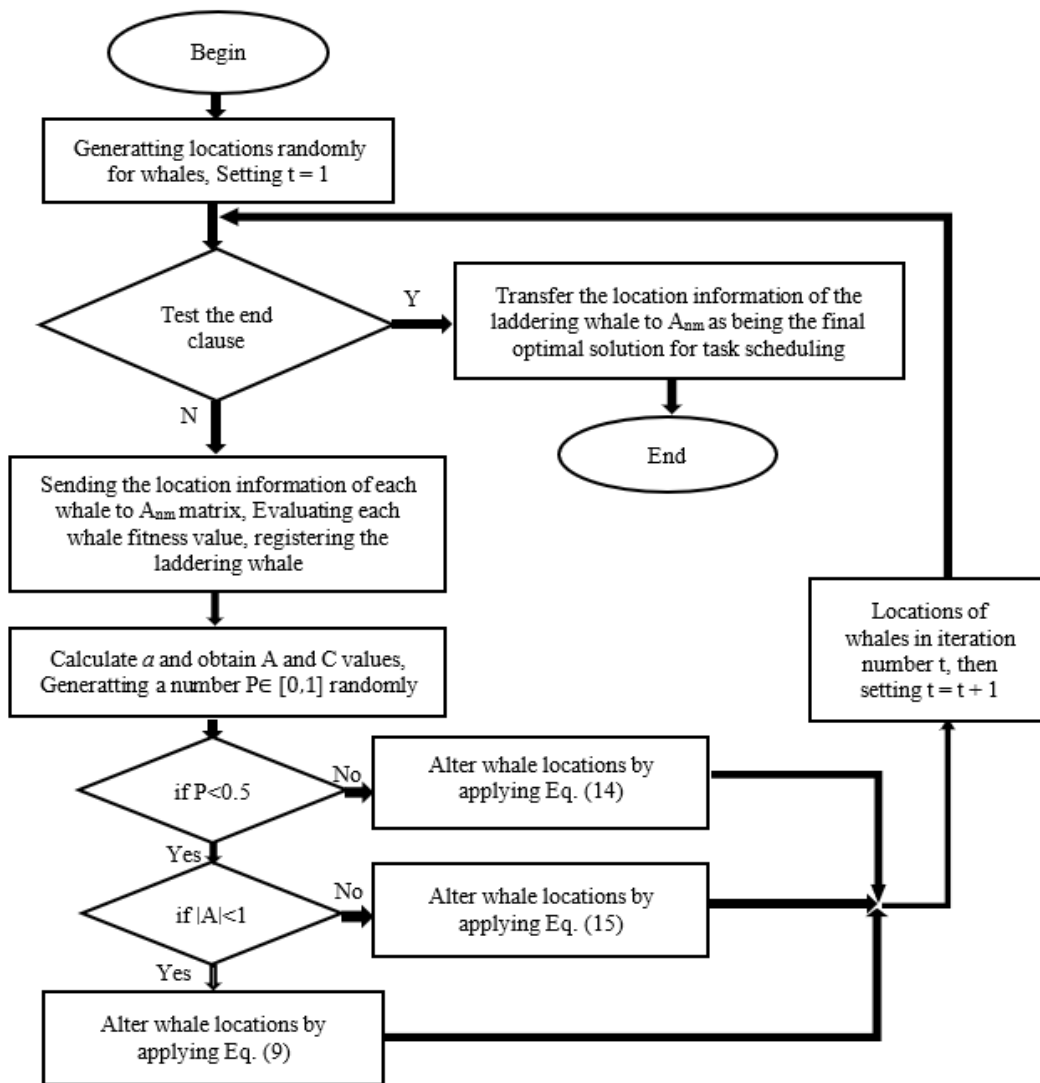


Figure 4. WOA process for cloud job scheduling

5. RESULT AND DISCUSSION

Due to the difficulty of accessing a real cloud computing environment, as creating such environments is very costly, there are many tools that facilitate the simulation process, providing environments similar to real cloud environments.

These tools allow for the implementation of algorithms, adjustment of main parameters, simulation environment configuration, and result visualization [26].

In this research, MATLAB is used to construct and apply WOA to job scheduling issue in cloud system context. It has been compared with other popular metaheuristic algorithms

previously applied at the same domain, such as Particle Swarm Algorithm (PSO) and Ant Colony Algorithm (ACO). The performance was evaluated through simulation experiments and compared with other algorithms in terms of, execution time cost, load cost, price cost, and the total cost F_{opt} .

The number of virtual machines has been set to 40 In order to, achieve a balance between improved performance and cost, the identification of this number of virtual devices is also ideal in terms of, the degree of the whale algorithm's computational complexity, and there are about 100 cloud jobs overall (which is normal for cloud platforms). Additionally, number of search agents has been specified as 50. Large number of iterations would clearly have a major impact on the execution of cloud tasks, so 100 iterations is the most that can be done. The characteristics and parameters related to tasks (Cloud Tasks) and virtual machines (Virtual Machines) used in the simulation experiments are shown in Table 2, where each choose a random number between the given range. The appropriate range for the parameters setup of tasks and virtual machines has been determined through practical experience and experimentation, taking into consideration several factors, such as the stability of the cloud system, resource efficiency and cost.

The experimental results provide detailed values for each of the following: execution time cost, load cost, price cost, and finally the total cost.

Table 2. Parameters setup of VMs and tasks

Parameter	Virtual Machine	Cloud Task
E CPU	[180,520]	[9,52]
S Memory	[80,520]	[40,105]
C Resource	[90,260]	[18,52]

5.1 Execution time cost

Algorithms performance in terms of, cost of execution time is displayed in Figure 5. Results showed that, whale algorithm has the lowest execution time cost compared to ACO and PSO algorithms, while looking for the optimal solution. This demonstrates whale algorithm's superior performance over the other algorithms in terms of, decreasing the CPU's energy usage as iterations number increases.

To be more specific, that may be observed that the values associated with the execution time cost are high in the initial iterations, which is natural considering the initial random scheduling. However, when more iterations are performed, these costs decrease. This demonstrates the effectiveness of the WOA in searching for optimal scheduling plans through the application of optimization rules, which enhance both local and global search operations to reach final optimal solution. In iteration 20 and beyond, the state becomes stable, where the values of the execution time cost approaches zero. This can be explained by the fact that the execution time cost, represented by the function $f1$ (which represents the power of CPU required for task relative to the CPU power of the virtual machine), becomes minimal for the most suitable virtual machine for the task as number of iterations increases. Therefore, the cost becomes close to zero (considering the CPU values taken by both the tasks and the virtual machines as shown in Table 2). This demonstrates the clear superiority of WOA over other algorithms in terms of, execution time cost.

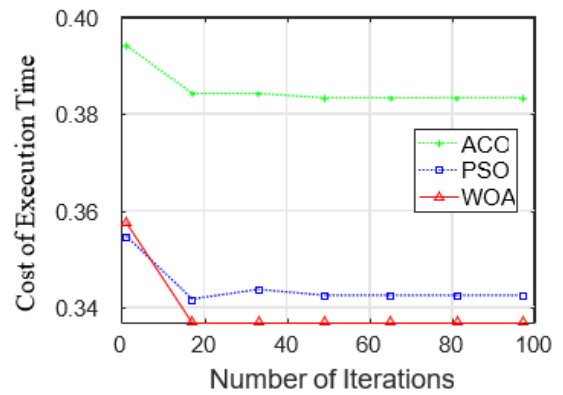


Figure 5. A comparison between the three algorithms' time costs after more iterations

5.2 Load cost

Figure 6 provides a detailed depiction of the load cost on the system for the three algorithms. It can be observed that all three algorithms exhibit efficient performance in reducing the load on the system as the number of iterations rises during the search for the best solution for the scheduling procedure. Interestingly, the PSO demonstrates a greater reduction in load cost compared to the whale algorithm. This is explained by balancing the final optimization goal function with the global optimum value (i.e., total cost), and the optimal value (i.e., the function for load costing). By giving up the local ideal value, which stands for the system load, the whale algorithm can obtain the global optimal value. This shows how well it can manage difficult optimization problems.

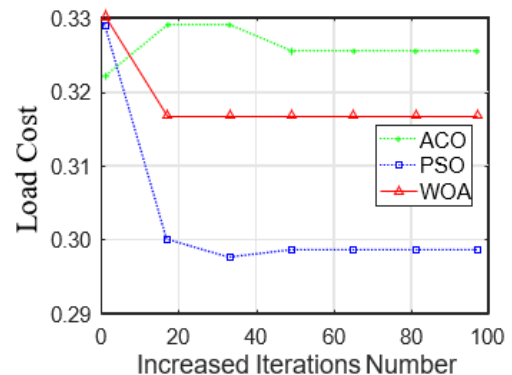


Figure 6. Comparison of algorithms load costs

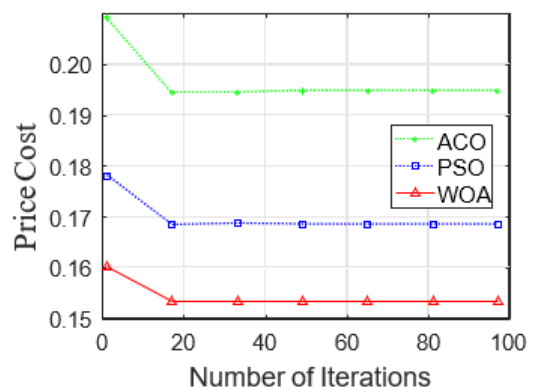


Figure 7. The three methods price costs at different numbers of iterations

5.3 Price cost

Figure 7 illustrates the price cost parameter, and it clearly shows that the whale algorithm outperforms the ACO and PSO. This shows how the whale algorithm may reduce system expenses as well as the economic cost of using resources, both of which increase the quality of cloud computing services.

5.4 Total cost

The values of the total cost, or the final optimization function, for each of the three techniques are displayed in Figure 8. The advantages provided by the three task scheduling approaches are illustrated by the observation that, for all algorithms, the overall cost lowers as number of iterations grows during the search for the best solution. Moreover, it is evident that, whale method performs better than the other algorithms, demonstrating its capacity to find the best work scheduling solution more quickly and precisely. Despite the relatively close performance of the WOA and PSO, the Whale algorithm consistently performs better.

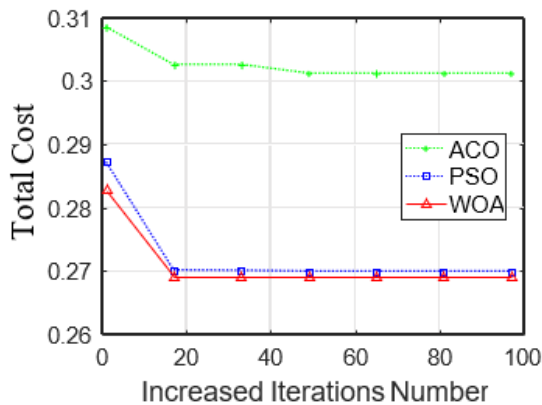


Figure 8. Comparison of total cost by raising the number of iterations of the three algorithms

It can be observed that, the function F_{opt} , refers to the optimization process's ultimate goal function, and ultimate representation of the three partial functions $f1$, $f2$, and $f3$. Therefore, it is the most important factor in evaluating the outcome of the optimization process. Despite the presence of a high partial value representing the load cost, the overall improvement is evident in the final result, emphasizing that the application of the WOA in cloud systems can offer a clearer more effective approach for cloud computing work scheduling, leading to enhanced Quality of Service in these systems.

The experimental results obtained in this research show the clear advantages provided by the application of the WOA on scheduling problems in cloud systems, where it provides an effective solution to the scheduling problem, including reducing execution time cost and economic cost compared to ACO and PSO. Although PSO provided better performance in reducing system load cost, the Whale Optimization Algorithm gives importance to the final objective function, which is the primary concern, where WOA provides the best solution in terms of, the total cost. Therefore, WOA remains the best approach.

Based on the execution parameters, cloud environment settings (such as the number of virtual machines, number of jobs, and detailed job parameters), and the results obtained, it

is evident that the application of the WOA is effective in typical cloud systems. This effectiveness is particularly notable in environments free from the complexities often associated with cloud operations, such as high task execution requirements and large volumes of tasks.

However, the application of WOA in complex cloud systems requires further developments and improvements, which will be part of our future work to achieve a dynamic WOA approach that considers the various requirements for task execution performance.

6. CONCLUSIONS

In this research, the latest metaheuristic algorithms (Whale Optimization Algorithm) were applied to solve the cloud-computing job scheduling issue. A paradigm for multi-objective task scheduling was used for arranging tasks to enhance the level of service quality. In comparison to other metaheuristic algorithms, WOA minimizes the overall cost, and minimizes the cost of execution time, and price, according to experimental data. Despite causing an increase in the load cost, the ultimate goal of improvement, which is the most important, is evident. This explains the ability of the WOA to effectively reach optimal scheduling plans.

The ultimate goal is to reach an effective, dynamic, and integrated methodology for task scheduling in cloud computing frameworks that achieves overall Quality of Service. This requires introducing improvements and developments to WOA to eliminate the limitations that directly affect the final scheduling results, such as convergence speed (in the case of large and complex scheduling problems) and scalability (a large number of tasks and complex requirements that need very high computational capabilities). Additionally, the re-tuning of the WOA parameters is required to reflect performance improvements.

In the future, it is possible to consider reducing the system load cost when applying the WOA by introducing some improvements to the search process (exploitation and exploration) for the optimal solution, in addition to addressing the three constraints of the WOA, which pose a future challenge, aiming to achieve an integrated approach that realizes optimal scheduling plans.

REFERENCES

- [1] Sato, M. (2009). Creating next generation cloud computing based network services and the contributions of social cloud operation support system (OSS) to society. In 2009 18th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises, Groningen, Netherlands, pp. 52-56. <http://doi.org/10.1109/WETICE.2009.18>
- [2] Shenai, S. (2012). Survey on scheduling issues in cloud computing. *Procedia Engineering*, 38: 2881-2888. <https://doi.org/10.1016/j.proeng.2012.06.337>
- [3] Sanaj, M.S., Prathap, P.J. (2021). An efficient approach to the map-reduce framework and genetic algorithm based Whale Optimization Algorithm for task scheduling in cloud computing environment. *Materials Today: Proceedings*, 37: 3199-3208. <https://doi.org/10.1016/j.matpr.2020.09.064>
- [4] Dorigo, M., Di Caro, G. (1999). Ant colony optimization:

- a new meta-heuristic. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, pp. 1470-1477. <https://doi.org/10.1109/CEC.1999.782657>
- [5] Fang, Y.Q., Xia, X., Ge, J.W. (2019). Cloud computing task scheduling algorithm based on improved genetic algorithm. In 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China, pp. 852-856. <https://doi.org/10.1109/ITNEC.2019.8728996>
- [6] Arunarani, A.R., Manjula, D., Sugumaran, V. (2019). Task scheduling techniques in cloud computing: A literature survey. *Future Generation Computer Systems*, 91: 407-415. <https://doi.org/10.1016/j.future.2018.09.014>
- [7] Elmougy, S., Sarhan, S., Joundy, M. (2017). A novel hybrid of shortest job first and round Robin with dynamic variable quantum time task scheduling technique. *Journal of Cloud computing*, 6: 12. <https://doi.org/10.1186/s13677-017-0085-0>
- [8] Zhang, P., Zhou, M. (2017). Dynamic cloud task scheduling based on a two-stage strategy. *IEEE Transactions on Automation Science and Engineering*, 15(2): 772-783. <http://doi.org/10.1109/TASE.2017.2693688>
- [9] Alworafi, M.A., Dhari, A., El-Booz, S.A., Nasr, A.A., Arpitha, A., Mallappa, S. (2019). An enhanced task scheduling in cloud computing based on hybrid approach. In *Data Analytics and Learning. Lecture Notes in Networks and Systems*, pp. 11-25. https://doi.org/10.1007/978-981-13-2514-4_2
- [10] Prasanna Kumar, K.R., Kousalya, K. (2020). Amelioration of task scheduling in cloud computing using crow search algorithm. *Neural Computing and Applications*, 32(10): 5901-5907. <https://doi.org/10.1007/s00521-019-04067-2>
- [11] Shafahi, Z., Yari, A. (2021). An efficient task scheduling in cloud computing based on ACO algorithm. In 2021 12th International Conference on Information and Knowledge Technology (IKT), Babol, Iran, Islamic Republic of, pp. 72-77. <http://doi.org/10.1109/IKT54664.2021.9685674>
- [12] Kumar, S.V., Nagaratna, M., Marrivada, L.H. (2022). Task scheduling in cloud computing using PSO algorithm. In *Smart Intelligent Computing and Applications, Volume 1: Proceedings of Fifth International Conference on Smart Computing and Informatics (SCI 2021)*, Singapore, pp. 541-550. https://doi.org/10.1007/978-981-16-9669-5_49
- [13] Anbarkhan, S.H., Rakrouki, M.A. (2023). An enhanced PSO algorithm for scheduling workflow tasks in cloud computing. *Electronics*, 12(12): 2580. <https://doi.org/10.3390/electronics12122580>
- [14] Odun-Ayo, I., Ajayi, O., Falade, A. (2018). Cloud computing and Quality of Service: Issues and developments. In Proceedings of the International Multi Conference of Engineers and Computer Scientists, IMECS 2018, Hong Kong, China, pp. 179-184.
- [15] Alhaidari, F., Balharith, T.Z. (2021). Enhanced round-robin algorithm in the cloud computing environment for optimal task scheduling. *Computers*, 10(5): 63. <https://doi.org/10.3390/computers10050063>
- [16] Kousalya, K., Kumar, K.R. (2016). QoS based task rescheduling in computational grid environment. *Asian Journal of Research in Social Sciences and Humanities*, 6(6): 1976-1992. <https://doi.org/10.5958/2249-7315.2016.00341.5>
- [17] El-Gamal, A.H., Mostafa, R.R., Hikal, N.A. (2020). Load balancing enhanced technique for static task scheduling in cloud computing environments. In *Internet of Things—Applications and Future: Proceedings of ITAF 2019*, pp. 411-430. https://doi.org/10.1007/978-981-15-3075-3_28
- [18] 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: 1-51. <https://doi.org/10.1007/s10115-017-1044-2>
- [19] Hu, Y., De Laat, C., Zhao, Z. (2019). Multi-objective container deployment on heterogeneous clusters. In 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Larnaca, Cyprus, pp. 592-599. <http://doi.org/10.1109/CCGRID.2019.00076>
- [20] Ramezani, F., Lu, J., Hussain, F. (2013). Task scheduling optimization in cloud computing applying multi-objective particle swarm optimization. In *Service-Oriented Computing: 11th International Conference, ICSOC 2013, Berlin, Germany*, pp. 237-251. https://doi.org/10.1007/978-3-642-45005-1_17
- [21] Li, W., Fan, Q., Dang, F., Jiang, Y., Wang, H., Li, S., Zhang, X. (2022). Multi-objective optimization of a task-scheduling algorithm for a secure cloud. *Information*, 13(2): 92. <https://doi.org/10.3390/info13020092>
- [22] 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*, 3: 2687-2699. <https://doi.org/10.1109/ACCESS.2015.2508940>
- [23] Mirjalili, S., Lewis, A. (2016). The Whale Optimization Algorithm. *Advances in Engineering Software*, 95: 51-67. <https://doi.org/10.1016/j.advengsoft.2016.01.008>
- [24] Nadimi-Shahraki, M.H., Zamani, H., Asghari Varzaneh, Z., Mirjalili, S. (2023). A systematic review of the Whale Optimization Algorithm: Theoretical foundation, improvements, and hybridizations. *Archives of Computational Methods in Engineering*, 30(7): 4113-4159. <https://doi.org/10.1007/s11831-023-09928-7>
- [25] Abdel-Basset, M., Manogaran, G., El-Shahat, D., Mirjalili, S. (2018). A hybrid Whale Optimization Algorithm based on local search strategy for the permutation flow shop scheduling problem. *Future Generation Computer Systems*, 85: 129-145. <https://doi.org/10.1016/j.future.2018.03.020>
- [26] Mansouri, N., Ghafari, R., Zade, B.M.H. (2020). Cloud computing simulators: A comprehensive review. *Simulation Modelling Practice and Theory*, 104: 102144. <https://doi.org/10.1016/j.simpat.2020.102144>