Saravanan *et al. Journal of Cloud Computing* (2023) 12:24
https://doi.org/10.1186/s13677-023-00401-1

Journal of Cloud Computing:
Advances, Systems and Applications

RESEARCH Open Access

Improved wild horse optimization with levy flight algorithm for effective task scheduling in cloud computing

G. Saravanan[1], S. Neelakandan[1,2*], P. Ezhumalai[3] and Sudhanshu Maurya[4]

Abstract

Cloud Computing, the efficiency of task scheduling is proportional to the effectiveness of users. The improved scheduling efficiency algorithm (also known as the improved Wild Horse Optimization, or IWHO) is proposed to address the problems of lengthy scheduling time, high-cost consumption, and high virtual machine load in cloud computing task scheduling. First, a cloud computing task scheduling and distribution model is built, with time, cost, and virtual machines as the primary factors. Second, a feasible plan for each whale individual corresponding to cloud computing task scheduling is to find the best whale individual, which is the best feasible plan; to better find the optimal individual, we use the inertial weight strategy for the Improved whale optimization algorithm to improve the local search ability and effectively prevent the algorithm from reaching premature convergence. To deliver services and access to shared resources, Cloud Computing (CC) employs a cloud service provider (CSP). In a CC context, task scheduling has a significant impact on resource utilization and overall system performance. It is a Nondeterministic Polynomial (NP)-hard problem that is solved using metaheuristic optimization techniques to improve the effectiveness of job scheduling in a CC environment. This incentive is used in this study to provide the Improved Wild Horse Optimization with Levy Flight Algorithm for Task Scheduling in cloud computing (IWHOLF-TSC) approach, which is an improved wild horse optimization with levy flight algorithm for cloud task scheduling. Task scheduling can be addressed in the cloud computing environment by utilizing some form of symmetry, which can achieve better resource optimization, such as load balancing and energy efficiency. The proposed IWHOLF-TSC technique constructs a multi-objective fitness function by reducing Makespan and maximizing resource utilization in the CC platform. The IWHOLF-TSC technique proposed combines the wild horse optimization (WHO) algorithm and the Levy flight theory (LF). The WHO algorithm is inspired by the social behaviours of wild horses. The IWHOLF-TSC approach's performance can be validated, and the results evaluated using a variety of methods. The simulation results revealed that the IWHOLF-TSC technique outperformed others in a variety of situations.

Keywords Task scheduling, Wild Horse Optimization (WHO), Cloud computing, Utilization of resources, Metaheuristic algorithms

*Correspondence:
S. Neelakandan
snksnk17@gmail.com
[1] Department of Computer Science and Engineering, Erode Sengunthar Engineering College, Erode- 638057, India
[2] Department of Computer Science and Engineering, R.M.K Engineering College, Chennai 601206, India
[3] Department of Computer Science and Engineering, R.M.D Engineering College, Chennai 601206, India
[4] School of Computing, Graphic Era Hill University, Bhimtal Campus, India

Introduction

Customers' access to diverse administrations and assets can be transformed using cloud computing. Clients can use the cloud to rapidly extend their capabilities without having to invest in licensing or foundations [1]. As time has passed, cloud computing has grown in popularity among clients as a means of obtaining administrations. Several task scheduling strategies have been proposed

© The Author(s) 2023. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

to enhance system performance by optimizing resource utilization. However, symmetry-based strategies have received little consideration. Given the widespread adoption of cloud computing technology, it is anticipated that symmetry will be employed to enhance cloud computing performance.

Cloud computing has the advantages of flexibility, constant quality, and less maintenance because the administrations and products are maintained by the outsider organization. These are the hypotheses that explain why the cloud is so prominent. Simplified terms for these cloud administrations include Software on Demand, Platform on Demand, and Infrastructure on Demand (IaaS) [2–4]. With cloud computing, multiple tasks can be carried out in the background at the same time. Server farm and board costs can be kept in check through the innovative use of virtualization and board robotization. In the age of cloud computing, making a reservation seems like a contradiction in terms. Ad-lobbing the asset utilization rate has the purpose of arriving at a perfect booking calculation Allocating diverse resources to the required errands is an NP-complete issue. Some Grid and Cloud system techniques have been implemented because of this factor. By employing meta-heuristics based on populations, it is possible to expand the size of the search area and yet achieve better outcomes [5–7]. Single-based meta-heuristic computations, rather than doing a thorough search, use different methods to arrive at the optimum answer for a random wellness task. As a result, utilizing population-based meta-heuristics to identify the best arrangement requires far less effort than using single-based meta-heuristics. Both calculations presented above have advantages and disadvantages. After then, a combination of meta-heuristics calculations is created by merging the two computations, which eliminates the negative marks in the separate calculations. With each recreation, the combined meta-heuristics calculation provides the best outcomes [8]. An alternative business model that is rapidly gaining traction around the world is the Internet-enabled business (e-Company). Internet-enabled businesses are transforming computing into a paradigm that consists of services that are commoditized and distributed in the same way that water, electricity, or gas are traditionally delivered. It doesn't matter where services are located or offered; users can use them according to their needs. Several computer paradigms have claimed to be able to deliver utility computing.

A cloud computing environment's fundamental purpose is to make the most efficient use of available computing resources. In the optimization process, scheduling methods are critical. As a result, user tasks must be scheduled using an efficient scheduling technique [9]. To reduce the total execution time, scheduling algorithms aim to distribute the workload across the available processors and maximize their efficiency. Many firms are migrating their infrastructure and operations to the cloud rather than retaining them on-premises, emphasizing the significance of installing a new CC solution. Cloud-based apps and services are becoming increasingly important [10]. As a result, the services' quality needed to be carefully addressed. Furthermore, the CC platform allows us to use many virtual resources for all the required tasks, rendering traditional and manual scheduling methods ineffective and demanding the creation of unique effective scheduling solutions.

However, task scheduling remains one of the major issues, potentially affecting performance, QoS, and user experience. As a result, the critical technical contribution in this paper, based on the idea is to introduce an enhanced solution with a novel genetic algorithm to exploit the collaboration between thin-thick clients and cloud network to optimise task scheduling of the processing system to deal with the issues, thereby improving QoS, user experience, and system reliability. Our proposal considers not only network contention but also the cost charged to cloud customers (CCs), as these two factors play important roles in meeting user expectations. Furthermore, the approach is experimentally evaluated and compared to others. The results show that our method is more efficient at task scheduling and more cost-effective than other approaches.

For cloud job scheduling, this study introduces the IWHOLF-TSC technique, which is an enhanced wild horse optimization with levy flight algorithm. The proposed IWHOLF-TSC technique reduces the CC platform's makespan and maximizes the utilization of the CC platform's resources to construct a multi-objective fitness function [11]. The suggested IWHOLF-TSC technique combines the wild horse optimization (WHO) algorithm with the Levy flight theory (LF). The social living traits of wild horses inspire the WHO algorithm. The performance validation of the IWHOLF-TSC approach can be done and the outcomes evaluated using a variety of methods.

This paper's contributions are as follows: (1) A metaheuristic optimization techniques for task scheduling is proposed, and IWHO is used to solve the entire problem; (2) an IWHO algorithm is proposed, which improves the convergence and accuracy of the WOA-based method, which improves the efficiency of task scheduling; and (3) describes the implementation process of the IWHOLF-TSC algorithm and compares it with the ACO, PSO, and WOA algorithms. The results of the experiments show that the algorithm works under a variety of task quantity conditions. The scheduling effect of down is superior.

The rest of this article is structured as follows. The related work is introduced in the second section. IWHOLF-TSC is introduced in the third section. The fourth section proposes the improved IWC implementation details, simulated the algorithm, and explained the scheduling effect. This article concludes with the fifth section.

Related work

In Pradeep, K., et al. [1], the Lion Advancement (LOA) and Opposition Based Learning (OBL) calculations are used in Crossbreed Oppositional Lion Enhancement Calculation (OLOA) With Cloud sim programming conditions, the given arrangement is recreated and displayed, and the obtained results show a significant improvement in execution over existing calculations such as Particle Swarm Optimization (PSO), Oppositional Dark Wolf Streamlining Agent (OGWO), and Genetic calculations (GA).

In Natesan, G. et al. [2], when compared to three recent planning calculations, GA-ETI reduced the time it took to execute work processes by 11–85 percent without increasing the cost. For a tough test and to integrate several upgrade objectives, GA-ETI shows the most efficient technique to building the optimal layer-scheduler for the framework of a work process supervisor. In Gobalakrishnan, N, et al. [3], an alternative to this approach is Particle Swarm Optimization (PSO), which makes use of both artificial and simulated ringing organisms in a symbiotic search (SASOS). Time spent by the executive team, implementation costs, communication costs, energy and resource consumption, epsilon restraint, and penalty characteristics are all comparable objectives.

In Casas, I., et al. [4], the multi-take interlace peak scheduling method (MIPSM) has been presented as a planning technique for quickly altering the content of resource loads. Three queues are created for CPU, CPU-intensive I/O memory capacity, and memory-intensive I/O. During the execution of planned operations, resource load peaks are dispersed. Using as minimal CPU, I/O, and memory as feasible is the purpose of each of the three queues in this system. In Zhou, J., et al. [5], to consider, the budget, budget, and performance optimization, I developed a resource cost model. To improve the quality of the solution, the augmented ant colony algorithm and feedback were utilized in a certain order. A check-up to reduce both runtime and fitness expenses, a hybrid glow-worm swarm optimization technique was created.

Nanjappan et al. [6] the proposed algorithm uses an ANFIS-BWO (Adaptive-Neuro-Fuzzy-Inference System-Black-Widow-Optimization) technique to allocate a proper VM for each task to reduce time delay. Another important goal for optimum consumption of cloud properties is resource scheduling. The BWO algorithm is used to generate an ideal solution set. The presented technique can allocate the VMs present on the cloud using the best scheduler schemes. The introduced technique's primary goal is to reduce computation time and cost while also minimising energy consumption for various tasks. Tong Zhou et al. [7] To design new reward functions to enhance the decision-making abilities of multiple reinforcement learning-based AI schedulers (RL). Using real-world case studies, the proposed methodology is evaluated and validated in a smart factory. Experiment results demonstrate that the new architecture for smart factories not only improves the learning and scheduling efficiency of multiple AI schedulers, but also effectively manages unanticipated events such as rush orders and machine breakdowns.

Mohammad Hasani Zade et al. [8] The suggested algorithm has two stages (i.e., meta-scheduler and local scheduler). The tasks are assigned to hosts in the meta-scheduler stage based on their priorities, completion dates, and host power. With the suggested Parallel Reinforcement Learning Caledonian Crow, the best mapping between tasks and virtual machines is discovered in the local-scheduler stage (PRLCC). The New Caledonian Crow Learning Algorithm (NCCLA), Reinforcement Learning (RL), and parallel strategy are all combined in the proposed PRLCC. Kaur & Kaur et al. [9]. To save cost and time developed a hybrid algorithm that combines heuristic and metaheuristic algorithms. To address the problem of task scheduling in cloud computing, which necessitates non-traditional optimization attitudes to achieve the optimal solution, the current paper proposes a hybrid multiple-objective approach called hybrid grey wolf and whale optimization (HGWWO) algorithms, which integrates two algorithms, namely, the grey wolf optimizer (GWO) and the whale optimization algorithm (WOA), with the goal of combining the advantages of each algorithm for miniaturisation.

Thekkepuryil et al. [10] the authors propose an improved version of the ant-lion optimization (ALO) algorithm that is crossbred with the popular particle swarm optimization (PSO) algorithm to improve system scheduling precision for cloud computing. A new security technique known as Data Encryption Standard (DES) is used, which encodes the information present in the cloud while scheduling is performed. The research's goal is to contribute an improved system scheduling framework that is more secure than existing frameworks. Improvement parameters are evaluated in terms of time, load, and cost. Shiau et al. [11] Cloud computing has been used to create cloud computing classrooms. Cloud computing student BI is unclear. Most researchers have compared few users' BI theories. This study tested, compared, and unified six theories: service quality (SQ), self-efficacy (SE), the motivational model (MM), the technology acceptance model (TAM), and extension of the Theory of Reasoned Action/ Theory of planned behavior (TRA/TPB) in cloud computing classrooms.

Palos-Sanchez, et al. [12]. studied Data Replication and Management, two important technologies for managing cloud data. These techniques ensure QoS for data operations (search, upload, download, replicate, etc.). Comparing and analysing techniques based on the above features. Researcher's analyses data replication techniques and cloud deployment of data-intensive apps. The paper's knowledge can be used to design and model cloud-based mechanisms and approaches. Medara et al. [13] The energy-efficient and reliability aware workflow task scheduling in cloud environment (EERS) algorithm is introduced in this paper, which conserves energy while making the most of the system's reliability. To begin, use a task-rank calculation programme to keep task dependencies intact. Following that, a task clustering algorithm is used to reduce communication costs, resulting in lower energy consumption.

T. Dillon et al. [14] presented the problems and difficulties with cloud computing. We explained how Cloud computing, Service-Oriented Computing, and Grid computing are related. We examined a few difficulties encountered when adopting cloud computing. After highlighting the interoperability problem, various solutions are discussed for various cloud service deployment models. Fan et al. [15] This study examines the short- and long-term competition between SaaS and SWS using a game theoretical approach. We examine both the long-term quality competition between the two firms as well as a model of price competition over one period. Software-as-a-service businesses can effectively differentiate their products by reducing the cost of system implementation by bundling software with services.

Agrawal et al. [16]. Cloud computing is a rapidly expanding and evolving platform. It offers its customers excellent services. Cloud computing development is significantly improving in terms of security in educational use. It provides numerous methods for accessing various resources' platform applications via on-demand web pages. This paper demonstrates various trending technologies, cloud computing features, and cloud computing security, as well as their application in education. Duraipandian. 2022 [17] Artificial Intelligence (AI) modernises today's society and paves the way for many digital applications to flourish. AI is important in everything from agriculture to space science. Improvements to electronic processors and other chips are also assisting AI in gaining computational efficiency. The design of micro electronic devices is allowing sensors to be moved to specific locations via Internet of Things (IoT) communication. The goal of this author is to examine the performance of AI methodologies in various applications to identify research gaps.

Palos-Sanchez et al., [18] examine the state of this technology from two angles: that of the European Union and, specifically, that of SMEs. The European Commission's strategic positions will then be evaluated considering the effects of adopting the cloud paradigm. Haag et al., [19] Identify five new research areas, including theoretical and practical phenomena, that describe the factors that influence the adoption of cloud services in organizations. These areas include information technology (IT)-related public threat appeals, trust in technology versus vendor trust, bring your own cloud, organizational identity transformation, and intelligent clusters of IT initiatives.

Ramezani et al., [20] an interval many-objective evolutionary algorithm (InMaOEA) is proposed for solving the cloud computing task scheduling problem. An interval many-objective cloud task scheduling optimization (I-MCTSO) model is developed by transforming uncertain factors into interval parameters. This model considers four optimization objectives: task completion rate, task make time, scheduling cost, and load balance. Mohammad Hasani et al., [21] a presentation designed to improve the performance of the Red Fox Optimization (RFO) algorithm. Initially, a Quasi-Opposition Based Learning method is used to generate the initial population, and a Levy flight method is employed to improve the exploratory ability of newly generated foxes. Two fuzzy control systems are used to achieve a balance between exploration and exploitation. Table 1 summarizes and compares the existing approaches.

In [5] the authors propose an improved version of the ant-lion optimization (ALO) algorithm that is crossbred with the popular particle swarm optimization (PSO) algorithm to improve system scheduling precision for cloud computing. A new security technique known as Data Encryption Standard (DES) is used, which encodes the information present in the cloud while scheduling is performed. The research's goal is to contribute an improved system scheduling framework that is more secure than existing frameworks. Improvement parameters are evaluated in terms of time, load, and cost.

Nanjappan et al. [6] The energy-efficient and reliability aware workflow task scheduling in cloud environment (EERS) algorithm is introduced in this paper, which conserves energy while making the most of the system's reliability. To begin, use a task-rank calculation programme to keep task dependencies intact. Following that, a task clustering algorithm is used to reduce communication costs, resulting in lower energy consumption.

Agrawal et al. [16]. Cloud computing is a rapidly expanding and evolving platform. It offers its customers excellent services. Cloud computing development is significantly improving in terms of security in educational use. It provides numerous methods for accessing various resources' platform applications via on-demand web pages. This paper demonstrates various trending technologies, cloud computing features, and cloud computing security, as well as their application in education.

Table 1 Summary of related works

Reference. No & Year	Methodology/Techniques	Features	Disadvantages
Ipsita Kar et.al. (2016) [22]	Genetic Algorithm and Darwin Theory	Global Search solution and self- managing scheme	Maximum Flowtime and resource utilization
Pan Yi.et.al (2016) [23]	Tabu Search based on techniques	Execution time and Throughput	Resource allocation time
X. -F. et.al (2018) [24]	Ant Colony Optimization	Randomization Techniques	Robustness and Effective Virtual Machine utility
M.Kaur.et.al (2016) [25]	Bacterial Foraging Algorithm	Makespan and Throughput	Minimum Flow time and Resource usage
S. Belgian (2014).[26]	Cat Swarm Optimization Algorithm	Optimal Resource utilization and Minimum iterations	In-secure Data access and throughput
Natesan.et.al (2019) [27]	grey wolf Algorithm	Randomization and parameter-based resource allocation	Multimodality support and convergence speed
Hariharan et. al. (2019) [28]	Whale and BAT optimization algorithm (WBAT)	Multi-objective job scheduling using hybridization	Complicated algorithm compared to other algorithms
K.Devi et. al. (2020) [29]	Deep Learning Based Cloud based Task Scheduling	Limits resource starvation and to guarantee fairness among the parties using the resources	As the number of cloud users increases, the scheduling becomes limited
Essam H et. al. (2021) [30]	Meta-heuristic task scheduling	Distribute complex tasks (cloudlets) to limited resources, within a reasonable time	The dilemmas of resources being underutilized (underloaded) and overutilized (overloaded)
Wanneng Shu et. al. (2021) [31]	Agile response task scheduling optimization	Explore the probability density function of the task request to avoid the timeout	Queue overflow happens, leads to network congestion
AK Reshmy et. al. (2019) [32]	Multilevel Fault-Tolerance Aware Scheduling Technique	Overcomes the real-time failure in the system	Mostly rely on the reactive scheme of checkpoint mechanism

Artificial Intelligence (AI) modernises today's society and paves the way for many digital applications to flourish. AI is important in everything from agriculture to space science. The advancement of cloud storage and wireless communication systems is assisting AI in achieving certain goals in the digital world. Improvements to electronic processors and other chips are also assisting AI in gaining computational efficiency. The design of micro electronic devices is allowing sensors to be moved to specific locations via Internet of Things (IoT) communication. The goal of this author is to examine the performance of AI methodologies in various applications in order to identify research gaps.

The proposed model

In this research, a new IWHOLF-TSC technique has been developed to effectually schedule tasks in CC environment. The proposed IWHOLF-TSC technique has derived a fitness function by minimizing make span and maximizing resource usage in the CC platform. The proposed IWHOLF-TSC technique integrates the concepts of WHO algorithm, which is stimulated from the social living characteristics of wild horses with LF. Figure 1 illustrates the system architecture of TS in CC.

Problem Formulation

The problem of TS from the cloud was determined as for scheduling, distributing, and assigning several various tasks to several VM efficiently and for performing every task that is able from minimum execution time [33]. The cloud system (CS) contains (N_{pm}) physical machines (PM), and all machines contain (N_{vm}) VM as demonstrated in Eq. (1).

$$CS = \left[PM_1, PM_2, \ldots, PM_i, \ldots, PM_{N_{pm}}\right] \tag{1}$$

where $PM_i, (i = 1, 2, \ldots, N_{pm})$ refers the PM carried out from the cloud and it could be written as:

$$PM = \left[VM_1, VM_2, \ldots, VM_k, \ldots, VM_{N_{vm}}\right] \tag{2}$$

where $VM_k, (k = 1, 2, \ldots, N_{vm})$ defines the k_{th} VM. N_{vm} signifies the amount of VM and VM_k refers the k_{th} VM devices from the cloud. The feature of VM_k was defined as:

$$VM_k = [SIDV_k, mips_k] \tag{3}$$

where id signifies the identifier amount of VM and $MIPS_k$ refers the report processing acceleration of VM by millions of instructions-per-seconds.

$$T = \left[Task_1, Task_1, \ldots, Task_i, \ldots, Task_{N_{tsk}}\right] \tag{4}$$

where N_{tsk} defines the number of tasks i projected by users. $Task_i$ implies the i_{th} task from the task series that is defined as:

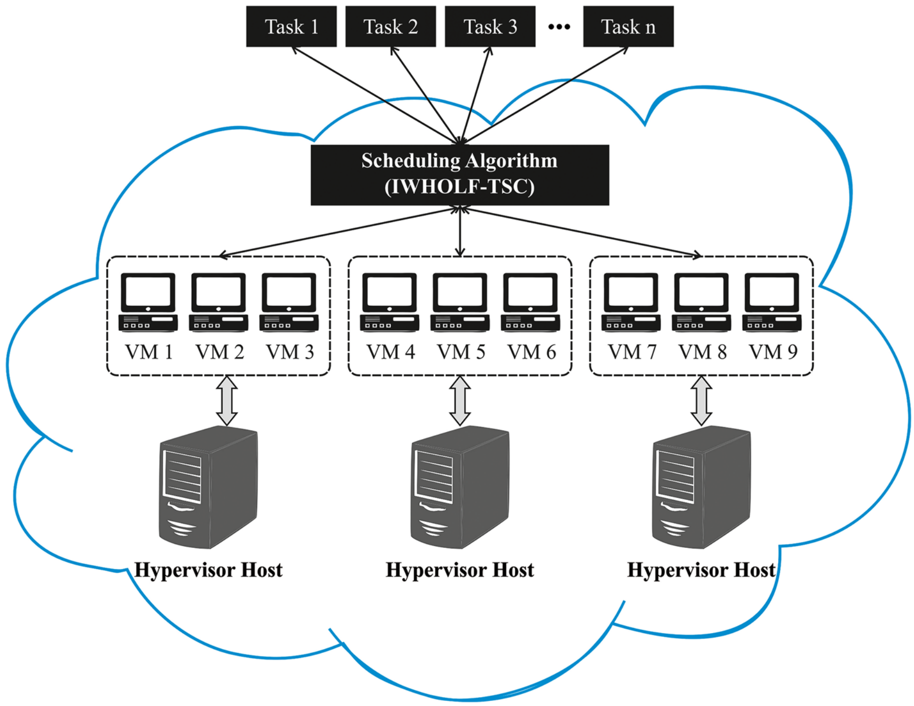

Fig. 1 System architecture of Task scheduling in CC

$$Task_i = [SIDT_i, Task - length_i, ECT_i, LI_i] \tag{5}$$

where $SIDT_i$ represents the identification number of j_{th} task and $task - length_i$ signifies the length of tasks [34]. Time ECT_i defines the predicted completion time to i_{th} task; LI_i refers the task preference from the number of tasks N_{tsk}. The Expected Complete Time (ECT) measure of size $N_{tsk} \times N_{vm}$ signifies the execution time required that carry out the tasks on all computing devices VM which is resolved by the subsequent matrix:

$$ECT = \begin{bmatrix} ECT_{1,1} & ECT_{1,2} & ECT_{1,3} & ECT_{1,N_{vm}} \\ ECT_{2,1} & ECT_{2,2} & ECT_{2,3} & ECT_{2,N_{vm}} \\ \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ ECT_{N_{t,k}},1 & ECT_{N_{tsk}},2 & \ldots & ECT_{N_{tsk},N_{vm}} \end{bmatrix} \tag{6}$$

Design of IWHOLF Algorithm

The WHO approach mathematically simulates and duplicates the social life performance of these wild horses naturally [35]. The horse usually lives from herd with stallion and several foals as well as mares. It can be demonstrated that variations of performances are containing mate and graze, pursue, dominate, command. In the 5 steps for WHO techniques are listed under Primary, an initial population was separated as to several groups. N refers the number of populations and G signifies the number of groups from this technique [36]. All groups have a leader (stallion), thus the number of stallions from this technique is equivalent G, and (NG) denotes the residual population (Foal and mare) were distributed similarly amongst these groups. The subsequent formula is presented for simulating the grazing performance:

$$X^j_{i,G} = 2Z\cos(2\pi RZ) \times \left(Stallion^j - X^j_{i,G}\right) + Stallion^j \tag{7}$$

where $X^j_{i,G}$ signifies the present place of foal/mare group members, *Stallion* defines the stallion place, R refers the uniform stochastic number in the range of -2 and 2, and Z implies the adaptive process computed in the subsequent formula:

$$P = \overrightarrow{R_1} < TDR; IDX = (P == 0); Z = R_2 \Theta IDX + \overrightarrow{R_3} \Theta(\sim IDX) \tag{8}$$

where P refers the vector containing *zero* to *one*, $\overrightarrow{R_1}$ and $\overrightarrow{R_3}$ implies the arbitrary number from the range of zero and one, R_2 represents the uniform arbitrary number from the range of zero and one, TDR stands for the adaptive parameter which begins with 1 and reduces still it attains 0 finally, the execution of technique based on subsequent formula:

$$TDR = 1 - it \times \left(\frac{1}{maxit}\right) \tag{9}$$

where *it* implies the existing iteration and *maxit* stands for the maximal number of iterations. For implementing the mate performance of horses, the foal drives in group i to temporary group but foal drives in group j to temporary group [37]. For simulating the mate performance of horses, the Crossover function of mean form is presented as:

$$X^{P}_{G,K} = Crossover\left(X^{q}_{G,i}, X^{Z}_{G,j}\right) i \neq j \neq k, p = q = end, Crossover = Mean \tag{10}$$

During the WHO technique, Stallions (group leaders) lead the group to water hole[38]. The Stallions compete to this water hole thus the domination group utilizes this water hole primarily afterward another group utilizes the water hole. The subsequent formula is mentioned that step of technique:

$$\overline{Stallion_{G_i}} = \begin{cases} 2Z\cos(2\pi RZ) \times (WH - Stallion_{G_i}) \\ +WHifR_3 > 0.5 \\ 2Z\cos(2\pi RZ) \times (WVH - Stallion_{G_i}) \\ -WHifR_3 \leq 0.5 \end{cases} \tag{11}$$

where $\overline{Stalllion_{G_i}}$ refers the next place of leaders. WH signifies the place of water hole. During the subsequent stage, leader is selected based on fitness. The leader place and relevant member are modifying dependent upon this formula:

$$\overline{Stallion_{G_i}} = \begin{cases} X_{G,i} if \cos t(X_{G,i}) < \cos t(Stallion_{G_i}) \\ Stallion_{G_i} if \cos t(X_{G,i}) > \cos t(Stallion_{G_i}) \end{cases} \tag{12}$$

The IWHOLF was dependent upon cuckoo search (CS) technique[39]. In the iteration of presented technique a novel solution was created utilizing the Levy flight as the subsequent formula:

$$X_{i,G} = X_{i,G} - \gamma(X_{i,G} - X_g) \oplus Levy(\lambda) = X_{i,G} + \frac{0.01u}{|v|^{1/\lambda}}(X_{i,G} - X_g) \tag{13}$$

where $X_{i,G}$ implies the i^{th} place of group member, γ represents the step scaling size, X_g stands for the global optimum solutions, $\oplus$ stands for the procedure of element-wise multiplication, λ denotes the Levy flight (LF) exponents, but u and v are determined as:

$$u \sim N\left(0, \sigma_u^2\right), v \sim N\left(0, \sigma_v^2\right) \tag{14}$$

The standard deviations σ_u and σ_v are formulated as:

$$\sigma_u = \left[\frac{\sin(\frac{\lambda\pi}{2})..\Gamma(1+\lambda)}{2^{(\lambda-1)}\lambda\Gamma(\frac{1+\lambda}{2})}\right]^{1/\lambda}, \sigma_v = 1 \tag{15}$$

where Γ refers the Gamma function, a novel candidate solution was created and was executed. The principal benefit of this enhancement was capability of the presented method for balancing global exploration and local exploitation.

Application of IWHOLF Algorithm for Task Scheduling

A major objective function of the IWHOLF technique is to reduce the makespan value by ordering the appropriate collection of tasks to be executed on VMs [40]. Since resource usage is related to makespan value, a higher usage value indicated that the CSP receives huge profit as shown in Algorithm.1.

$$ECT_{ik} = \frac{task - length}{mips_k} \tag{16}$$

$$Ru = \frac{\sum_{i=1}^{m} CT_i}{makespan \times m} \tag{17}$$

```
Create VMs in datacenter and submit task list to VMs
Prepare List of VMs and Task
Initialize ECT of ith tasks on kth VM
      where k = 1, 2, 3,..., N_VM, i = 1, 2, 3,..., N_tsk,
calculate ECT_ik = task-length/mips_k
Schedule List of tasks and put task in the queue
for all i <- 1 to size of scheduled task list do
    for all j<- 1 to size of non-scheduled task list do
      if task j scheduled before task i then
         put task j into scheduled queue
      else
         put task i into scheduled queue
      end if
    end for
  end for
arrange the list of tasks according to scheduled order
arrange the VMs received task list in descending order.
for all i <- 0 to size of scheduled task list do
  if j>=0 then
     Execute the task_i to the VMs J++
  if j== Count of VMs then
     J=0
  end if
end if
end for
```

Algorithm. 1 Task Scheduling

where $k = 1, 2, 3,..., N_{VM}$, i = 1, 2, 3,..., N_{tsk}, and ECT_{ik} means required execution time of ith tasks on kth VM. N_{VM} represents the total VM count and N_{tsk} represents task count. The integrated fitness value with multi-objective function of all collections is determined by the use of Eq. (13) that represents the evolution power of the organisms.

$$F = \left(max\left\{ECT_{ik}\right\}\&min\left\{Ru_k\right\}\right), \forall \in \left[1, N_{tsk}\right] mappedtokthVM \tag{18}$$

Generally, the demanded tasks are planned to free VMs and the tasks are attended depending upon the order. The major intention of task scheduling over VMs is the way of attaining high usage of VMs along with minimal makespan values[41]. The Expected Time to Compute (ETC) of provide task to be listed on every VM can be used by the IWHOLF-TSC technique for scheduling process.

Experimental validation

To further validate the algorithm's task scheduling effect in cloud computing, the IWC algorithm is compared to the ACO, PSO, and WOA algorithms. Table 1 lists the parameters required by the algorithm. Choose a CPU Core i3, memory 4 G DDR3, a hard disc capacity of 1000 G, Windows 7, and MATLAB 2012. The experiment is divided into small-scale and large-scale cloud computing tasks. The comparison indicators are cost value, time value, and memory load value. To explain the effect of better scheduling, this paper sets the number of small-scale tasks to [0, 1000] and the number of large-scale tasks to [1000, 10000].

To demonstrate the algorithm's efficiency in cloud computing task scheduling, the ant colony algorithm (ACO), particle swarm optimization (PSO), and whale optimization algorithm (WOA) were chosen from the classical algorithms and compared with the algorithm proposed herein for cloud computing task scheduling. The CloudSim simulation platform was then used to simulate the cloud computing environment. Table 2 shows the main parameters required by the algorithm in this case. When combined with the characteristics of the tasks in cloud computing, the tasks were divided into small-scale tasks and large-scale tasks, which were compared in the QoS indicators based on time and cost.

A brief makespan (MKS) analysis of the IWHOLF-TSC technique is compared with recent methods under small tasks are provided in Table 3 and Fig. 2. The results referred that the IWHOLF-TSC method has obtainable lower MKS under all tasks. For instance, with 100 tasks, the IWHOLF-TSC system has provided minimal MKS of 58 whereas the WOA, MSA, ALO, and MALO approaches have reached increased MKS of 94, 88, 83, and 66 correspondingly. In addition, with 500 tasks, the IWHOLF-TSC method has demonstrated lower MKS of 382 whereas the WOA, MSA, ALO, and MALO techniques have able higher MKS of 480, 458, 439, and 425 correspondingly [74–77]. At last, with 1000 tasks, the IWHOLF-TSC method has provided decreased MKS of 776 whereas the WOA, MSA, ALO, and MALO techniques have depicted improved MKS of 961, 931, 904, and 858 correspondingly.

Table 2 Simulation environments

Parameters	Value
Simulation Software Tool	CloudSim Software Version 4.0
Host Machine	Intel i5 6300U CPU @ 2.4 GHz
Host Machine Memory Capacity	20 GB
No. of Virtual Machine	32
No. of Cloudlet	1024

Table 3 MKS Analysis of IWHOLF-TSC model under small tasks

Makespan (Small Tasks)					
No. of Tasks	**WOA**	**MSA**	**ALO**	**MALO**	**IWHOLF-TSC**
100	94	88	83	66	58
200	153	137	123	113	96
300	292	273	249	229	191
400	379	360	333	306	265
500	480	458	439	425	382
600	618	586	539	520	480
700	716	678	654	629	572
800	817	781	724	705	659
900	896	871	814	787	730
1000	961	931	904	858	776

A detailed MKS analysis of the IWHOLF-TSC approach is compared with recent algorithms under large tasks are given in Table 4 and Fig. 3. The results demonstrated that the IWHOLF-TSC method has offered reduced MKS under all tasks. For instance, with 1100 tasks, the IWHOLF-TSC method has provided minimal MKS of 1054 but the WOA, MSA, ALO, and MALO techniques have attained higher MKS of 1172, 1124, 1140, and 1086 correspondingly. Followed by, with 1500 tasks, the IWHOLF-TSC method has outperformed lower MKS of 1406 whereas the WOA, MSA, ALO, and MALO algorithms have accomplished higher MKS of 1539, 1476, 1460, and 1428 correspondingly. Besides, with 2000 tasks, the IWHOLF-TSC methodology has provided lower MKS of 1793 whereas the WOA, MSA, ALO, and MALO systems have portrayed increased MKS of 1999, 1989, 1970, and 1901 respectively.

A degree of imbalance (DOI) analysis of the IWHOLF-TSC method is compared with recent methods under numerous tasks in Table 5 and Fig. 4. The results indicated that the IWHOLF-TSC method has offered reduced DOI under all tasks [78–82]. For instance, with 100 tasks, the IWHOLF-TSC method has provided

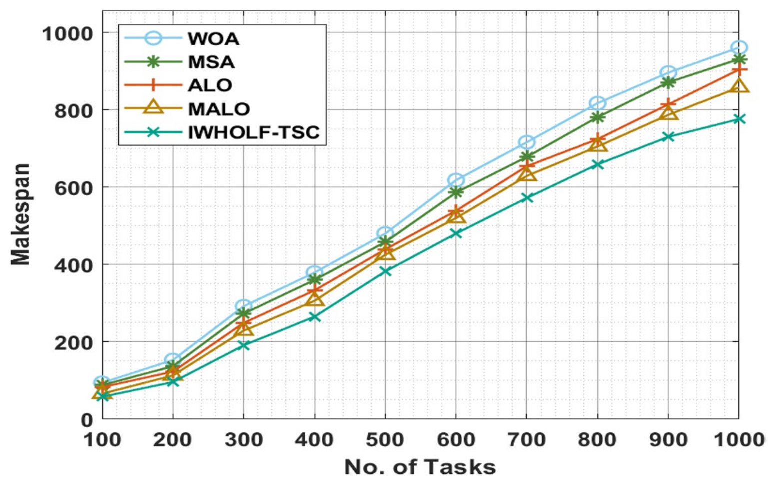

Fig. 2 Comparative MKS Analysis of IWHOLF-TSC model under small tasks

Table 4 MKS Analysis of IWHOLF-TSC model under large tasks

Makespan (Large Tasks)					
No. of Tasks	**WOA**	**MSA**	**ALO**	**MALO**	**IWHOLF-TSC**
1100	1172	1124	1140	1086	1054
1200	1229	1191	1172	1134	1099
1300	1365	1340	1321	1267	1203
1400	1432	1413	1381	1349	1292
1500	1539	1476	1460	1428	1406
1600	1692	1647	1609	1530	1492
1700	1755	1720	1688	1622	1577
1800	1875	1825	1790	1726	1692
1900	1986	1901	1863	1821	1749
2000	1999	1989	1970	1901	1793

minimal DOI of 0.843 whereas the WOA, MSA, ALO, and MALO techniques [83–88] have reached maximum DOI of 1.628, 1.534, 1.327, and 0.965 respectively.

Moreover, with 500 tasks, the IWHOLF-TSC method has demonstrated lower DOI of 0.866 whereas the WOA, MSA, ALO, and MALO techniques have accomplished higher DOI of 1.853, 1.745, 1.360, and 1.049 respectively. Furthermore, with 1000 tasks, the IWHOLF-TSC method has provided decreased DOI of 0.885 whereas the WOA, MSA, ALO, and MALO techniques have depicted improved DOI of 2.450, 2.182, 1.623, and 1.280 respectively.

Table 6 and Fig. 5 provide a fitness function analysis of the WOH and IWHOLF techniques under distinct iterations. The results indicated that the IWHOLF technique

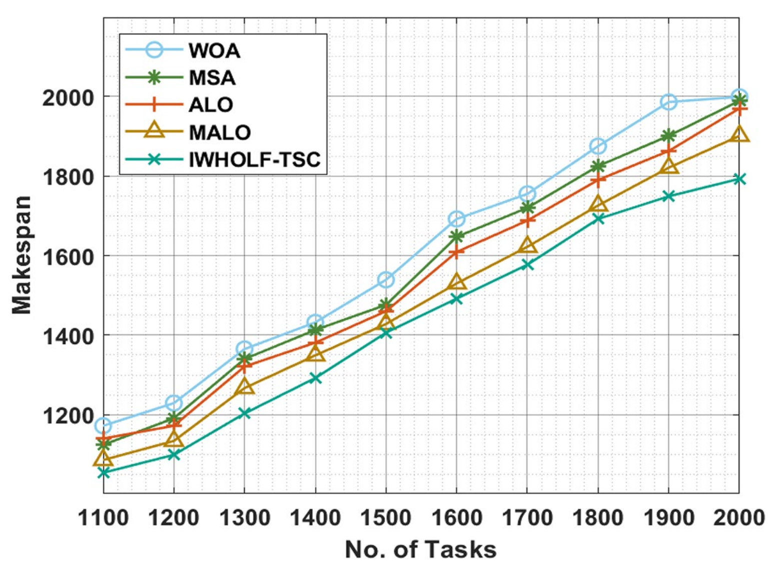

Fig. 3 Comparative MKS Analysis of IWHOLF-TSC model under large tasks

Table 5 DOI Analysis of IWHOLF-TSC model under diverse tasks

Degree of imbalance					
No. of Tasks	**WOA**	**MSA**	**ALO**	**MALO**	**IWHOLF-TSC**
100	1.628	1.534	1.327	0.965	0.843
200	1.369	1.294	1.261	1.002	0.866
300	1.364	1.327	1.214	1.026	0.857
400	1.745	1.613	1.115	0.970	0.852
500	1.853	1.745	1.360	1.049	0.866
600	1.858	1.712	1.463	1.045	0.871
700	2.163	1.764	1.618	1.021	0.880
800	2.309	1.971	1.529	1.092	0.857
900	2.445	2.145	1.576	1.148	0.918
1000	2.450	2.182	1.623	1.280	0.885

Table 6 Fitness Function Analysis of IWHOLF with WOH algorithms

Fitness Function		
No. of Iterations	**WOH**	**IWHOLF**
0	206.52	209.13
100	197.38	188.46
200	108.76	95.86
300	96.04	85.93
400	97.23	78.38
500	96.84	77.58
600	96.44	77.58
700	86.50	77.58
800	86.50	77.58
900	86.50	77.58
1000	86.50	77.58

has resulted in effectual outcomes with optimal fitness values under all iterations. For instance, with 100 iterations, the IWHOLF technique has resulted in lower fitness value of 188.46 whereas the WHO has offered slightly increased fitness value of 197.38. In addition, with 500 iterations, the IWHOLF technique has attained reduced lower fitness value of 77.58 whereas the WHO has reached considerably enhanced fitness value of 96.84. Concurrently, with 1000 iterations, the IWHOLF technique has resulted in lower fitness value of 77.58 whereas the WHO has offered slightly increased fitness value of 86.50.

Finally, a computation time (CT) analysis of the IWHOLF-TSC method is compared with recent methods under numerous tasks in Table 7 and Fig. 6. By observing the values, it is evident that the IWHOLF-TSC method has attained minimal CT under all tasks. For instance, with 100 tasks, the IWHOLF-TSC method has offered lower CT of 19.05 min whereas the WOA, MSA, ALO, and MALO techniques have obtained higher CT of 155.70 min, 111.33 min, 72.28 min, and 51 min respectively.

Moreover, with 500 tasks, the IWHOLF-TSC method has reached minimal CT of 320.75 min whereas the WOA, MSA, ALO, and MALO techniques have accomplished maximum CT of 446.77 min, 409.48 min, 386.42 min, and 365.12 min respectively. Furthermore, with 1000 tasks, the IWHOLF-TSC method has reached reduced CT of 489.35 min whereas the WOA, MSA, ALO, and MALO techniques have exhibited increased CT of 636.67 min, 602.93 min, 624.23 min, and 572.77 min respectively. The above-mentioned tables and figures ensured the betterment of the IWHOLF-TSC technique over the other methods.

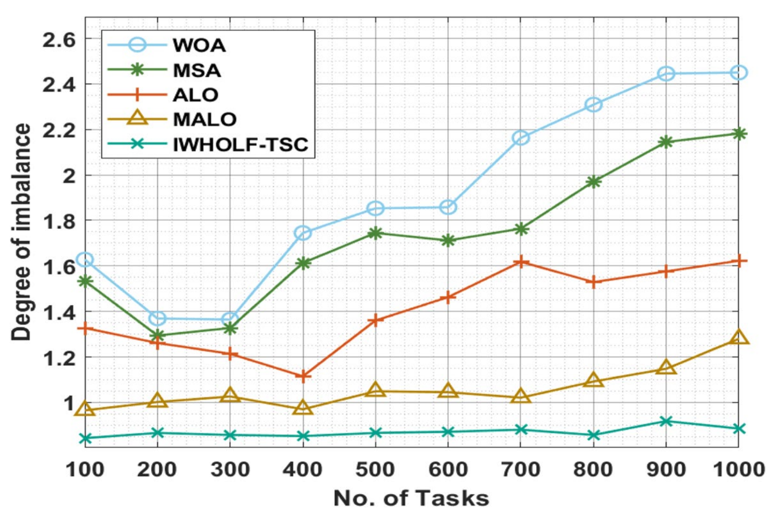

Fig. 4 Comparative DOI Analysis of IWHOLF-TSC with existing models

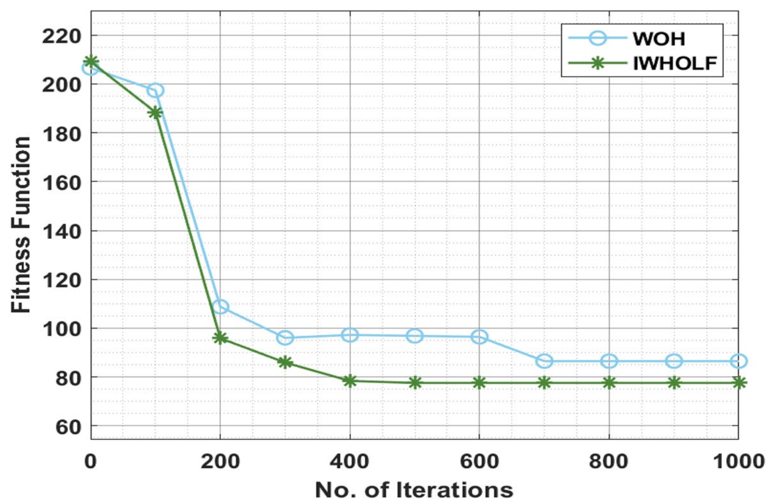

Fig. 5 Comparative Fitness Function Analysis of IWHOLF with WOH algorithms

Table 7 CT Analysis of IWHOLF-TSC model under varying tasks

Computation Time (min)					
No. of Tasks	**WOA**	**MSA**	**ALO**	**MALO**	**IWHOLF-TSC**
100	155.70	111.33	72.28	51.00	19.05
200	155.70	130.85	98.92	81.17	42.12
300	281.72	249.77	230.25	216.05	146.83
400	311.88	287.03	265.73	260.42	210.72
500	446.77	409.48	386.42	365.12	320.75
600	480.48	444.98	420.13	398.85	370.45
700	505.33	475.15	455.63	437.88	411.27
800	574.55	537.27	521.30	500.00	464.52
900	617.13	578.10	588.73	544.37	480.48
1000	636.67	602.93	624.23	572.77	489.35

Figure 7 and Table 8 illustrate a comparative success rate examination of the IWHOLF-TSC approach with other existing methods. The figure shows that the cloud computing approach has resulted in higher performance with success rate. For example, with no of tasks 100, the success rate value is 85.478% for IWHOLF-TSC, whereas the WOA, MSA, ALO, and MALO models have obtained success rate of 77.345%, 73.768%, 61.526%, and 65.026%, respectively. However, the IWHOLF-TSC model has shown maximum performance with different data set size. Similarly, under 1000 tasks, the success rate value of IWHOLF-TSC is 93.775%, while it is 83.879%, 75.943%, 64.837%, and 68.937% for WOA, MSA, ALO, and MALO models, respectively.

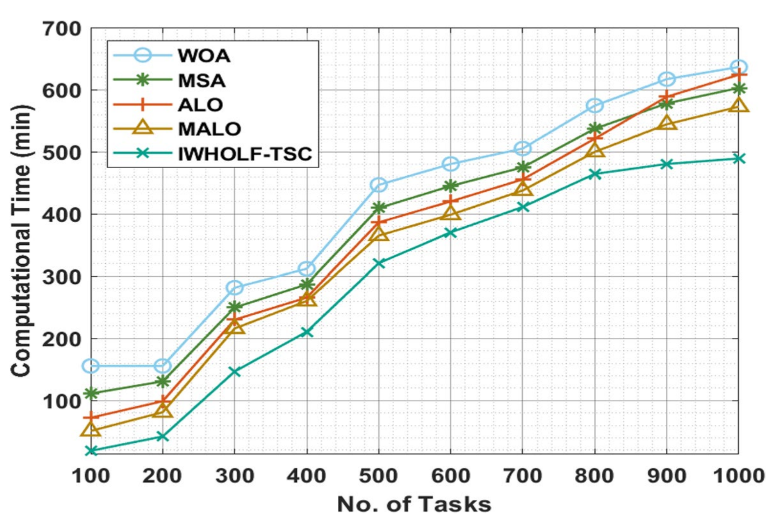

Fig. 6 Comparative CT Analysis of IWHOLF-TSC model

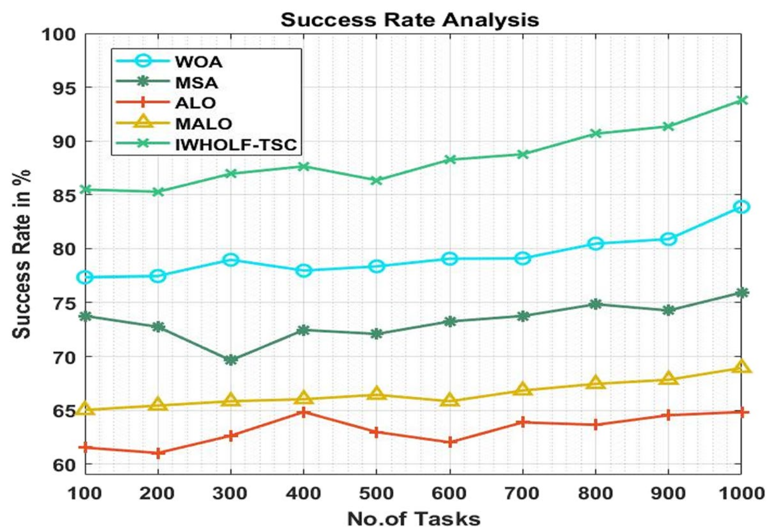

Fig. 7 Success rate analysis of IWHOLF-TSC model under varying tasks

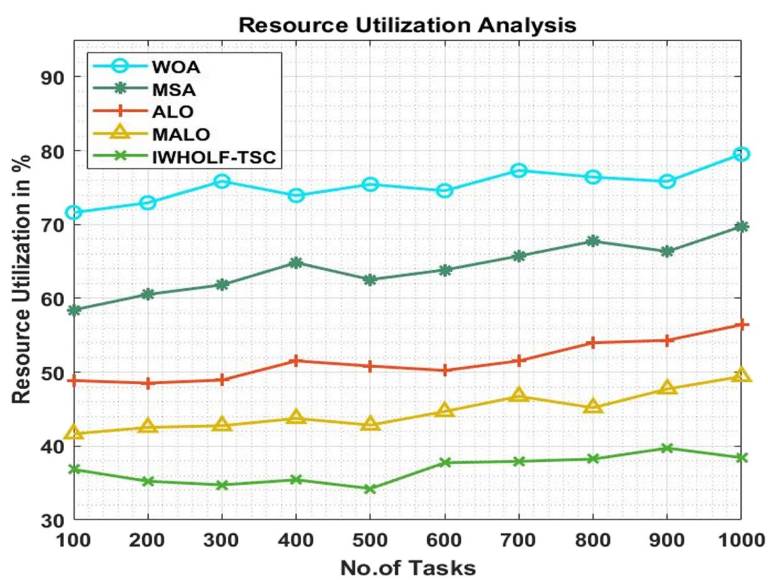

Fig. 8 Resource utilization analysis of IWHOLF-TSC model under varying tasks

Table 9 and Fig. 8 explain the resource utilization of the IWHOLF-TSC method with other existing techniques. The data clearly explains that the proposed method has the least resource utilization compared to the other methods in all aspects. For example, with 100 tasks, the proposed method has a resource utilization of 36.837%, while it is 71.623%, 58.436%, 48.873%, and 41.653% for WOA, MSA, ALO, and MALO, respectively. The IWHOLF-TSC method has greater performance with less resource utilization. Similarly, with 1000 tasks, the proposed method has 38.434% of resource utilization whereas the methods for WOA, MSA, ALO, and MALO have resource utilization of 79.526%, 69.733%, 56.432%, and 49.432%, respectively.

Conclusion

In this article, a new IWHOLF-TSC technique for effectively scheduling tasks in a CC environment is presented. The purpose of this article is to introduce an IWHOLF-TSC based task scheduling method in cloud computing task scheduling, with the goal of improving the effect of cloud computing task scheduling. Two optimization strategies based on the IWHOLF-TSC algorithm are proposed to further improve scheduling

Table 8 Success rate analysis of IWHOLF-TSC model under varying tasks

Success Rate					
No. of Tasks	**WOA**	**MSA**	**ALO**	**MALO**	**IWHOLF-TSC**
100	77.345	73.768	61.526	65.026	85.478
200	77.456	72.754	61.028	65.436	85.278
300	78.954	69.637	62.633	65.836	86.953
400	77.954	72.463	64.837	66.028	87.643
500	78.345	72.095	62.973	66.435	86.346
600	79.054	73.256	62.017	65.833	88.257
700	79.086	73.764	63.873	66.836	88.764
800	80.456	74.854	63.647	67.452	90.674
900	80.876	74.276	64.536	67.836	91.356
1000	83.879	75.943	64.837	68.937	93.775

Table 9 Resource utilization analysis of IWHOLF-TSC model under varying tasks

Resource Utilization					
No. of Tasks	**WOA**	**MSA**	**ALO**	**MALO**	**IWHOLF-TSC**
100	71.623	58.436	48.873	41.653	36.837
200	72.937	60.537	48.532	42.536	35.246
300	75.856	61.826	48.937	42.763	34.746
400	73.923	64.836	51.534	43.744	35.435
500	75.435	62.548	50.833	42.844	34.243
600	74.569	63.846	50.243	44.684	37.736
700	77.321	65.735	51.537	46.736	37.938
800	76.424	67.736	53.984	45.192	38.243
900	75.826	66.347	54.323	47.736	39.732
1000	79.526	69.733	56.432	49.432	38.434

performance based on the IWHOLF-TSC algorithm. When compared to some commonly used metaheuristic algorithms, the experimental results show that it can be used in system load and system resource utilisation. Cloud computing systems' cost-effectiveness has greatly improved. The proposed IWHOLF-TSC technique derives a fitness function in the CC platform by minimising Makespan and maximising resource usage. The proposed IWHOLF-TSC technique incorporates WHO algorithm concepts that are stimulated by the social living characteristics of wild horses with LF. The performance validation of the IWHOLF-TSC technique can be carried out and the results evaluated using a variety of metrics. The IWHOLF-TSC method achieved a lower MKS of 382 while the WOA, MSA, ALO, and MALO techniques achieved higher MKS of 480, 458, 439, and 425, respectively. The IWHOLF-TSC method yielded a low MKS of 1054, whereas the WOA, MSA, ALO, and MALO techniques yielded higher MKS of 1172, 1124, 1140, and 1086, respectively. In the simulation experiment, IWHOLF-TSC was pitted against the ant colony algorithm, the particle swarm algorithm, and the whale optimization algorithm across a variety of tasks. According to the results, the IWHOLF-TSC algorithm performs well in terms of task scheduling time, scheduling cost, and virtual machine. The application is used to schedule cloud computing tasks. Resource allocation and clustering processes for IoT assisted cloud environments can be designed as part of the future scope.

Authors' contributions
Conceptualization, G. Saravanan and Neelakandan. S.;validation, Neelakandan.S. and P. Ezhumalai; formal analysis, P. Ezhumalai and G. Saravanan; data curation, G. Saravanan, and Sudhanshu Maurya; writing—original draft preparation, Neelakandan S and G. Saravanan; writing—review and editing, P. Ezhumalai and Sudhanshu Maurya. The authors read and approved the final manuscript.

Funding
The authors received no specific funding for this study.

Availability of data and materials
Not Applicable.

Declarations

Competing interests
The authors declare no competing interests.

Ethics Approval and consent to participate
We confirm that our research does not involves a survey asking real human participants to give opinions, or animals data to make justifications.

Competing interest
The authors declare no conflict of interest.

Received: 24 November 2022 Accepted: 1 February 2023
Published online: 21 February 2023

References

1. Pradeep K, Ali LJ, Gobalakrishnan N, Raman CJ, Manikandan N (2022) CWOA: Hybrid approach for task scheduling in cloud environment. Comput J 65:1860–1873. https://doi.org/10.1093/comjnl/bxab028
2. Natesan G, Chokkalingam A. An improved grey wolf optimization algorithm based task scheduling in cloud computing environment. Int. Arab J. Inf. Technol. 2020, 17, 73–81. https://doi.org/10.34028/iajit/17/1/9
3. Gobalakrishnan N, Arun C (2018) A new multi-objective optimal programming model for task scheduling using genetic gray wolf optimization in cloud computing. Comput J 61:1523–1536. https://doi.org/10.1093/comjnl/bxy009
4. Casas I, Taheri J, Ranjan R, Wang L, Zomaya AY (2018) GA-ETI: an enhanced genetic algorithm for the scheduling of scientific workflows in cloud environments. J Comput Sci 26:318–331. https://doi.org/10.1016/j.jocs.2016.08.007

5. Zhou J, Dong S (2018) Hybrid glowworm swarm optimization for task scheduling in the cloud environment. Eng Optim 50:949–964. https://doi.org/10.1080/0305215X.2017.1361418
6. Nanjappan M, Natesan G, Krishnadoss P (2021) An adaptive neuro-fuzzy inference system and black widow optimization approach for optimal resource utilization and task scheduling in a cloud environment. Wireless Pers Commun 121(3):1891–1916
7. Zhou T, Tang D, Zhu H, Zhang Z (2021) Multi-agent reinforcement learning for online scheduling in smart factories. Robot Comput Integrat Manufact 72
8. Zade BM, Mansouri N, Javidi MM. A two-stage scheduler based on New Caledonian Crow Learning Algorithm and reinforcement learning strategy for cloud environment. J Network Comput App. 2022 202:103385.
9. Kaur M (2016) Elitist multi-objective bacterial foraging evolutionary algorithm for multi-criteria based grid scheduling problem. International Conference on Internet of Things and Applications (IOTA) 431–436:2016. https://doi.org/10.1109/IOTA.2016.7562767
10. Thekkepuryil JKV, Suseelan DP, Keerikkattil PM (2021) An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment. Clust Comput 24(3):2367–2384
11. Shiau WL, Chau PYK (2016) Understanding behavioral intention to use a cloud computing classroom: A multiplemodel comparison approach. Inf Manage 53:355–365
12. Palos-Sanchez PR, Robina-Ramirez R, Velicia-Martin F (2019) What role does corporate governance play in the intention to use cloud computing technology. Symmetry 11:1253
13. Medara R, Singh SR (2021) Energy efficient and reliability aware workflow task scheduling in cloud environment. Wireless Pers Commun 119(2):1301–1320
14. Dillon T, Wu C, Chang E. Cloud computing: issues and challenges. In Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA '10) 2010, Perth, Australia, 27–33.
15. Fan M, Kumar S, Whinston AB (2019) Short-term and long-term competition between providers of shrink-wrap software and software as a service. Eur J Oper Res 196:661–671
16. Agrawal R, Bansal S. (2022). Cloud Computing: Security with Educational Usage. J IoT Soc Mobile Analytics Cloud 4(3): 165–173.
17. Duraipandian M (2022) Review on Artificial Intelligence and its Implementations in Digital Ersa. J Inform Technol Digit World. 4(2):84–94
18. Palos-Sanchez PR (2017) Drivers and Barriers of the Cloud Computing in SMEs: the Position of the European Union. Harvard Deusto Business Research 6:116–132
19. Haag S, Eckhardt A (2014) Organizational cloud service adoption: a scient metric and content-based literature analysis. J Bus Econ 84:407–440
20. Ramezani F, Lu J, Taheri J (2015) Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments. World Wide Web 18:1737–1757
21. Zade BM, Mansouri N (2022) Improved red fox optimizer with fuzzy theory and game theory for task scheduling in cloud environment. J Comput Sci. 63
22. Kar I, Parida RNR, Das H. Energy aware scheduling using genetic algorithm in cloud data centers. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT) 2016, 3545–3550. doi: https://doi.org/10.1109/ICEEOT.2016.7755364.
23. Yi P, Ding H, Ramamurthy B (2016) Budget-Optimized Network-Aware Joint Resource Allocation in Grids/Clouds Over Optical Networks. J Lightwave Technol. 34:3890–3900
24. Liu XF, Zhan ZH, Deng JD, Li Y, Gu T, Zhang J. (2018). An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Transact Evol Comput. 22(1):113–128, https://doi.org/10.1109/TEVC.2016.2623803.
25. Kaur N, Chhabra A. (2016). Analytical review of three latest nature inspired algorithms for scheduling in clouds. International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT). 3296–3300. https://doi.org/10.1109/ICEEOT.2016.7755315.
26. Bilgaiyan S, Sagnika S, Das M. (2014). Workflow scheduling in cloud computing environment using Cat Swarm Optimization. IEEE International Advance Computing Conference (IACC). 680–685, doi: https://doi.org/10.1109/IAdCC.2014.6779406.
27. Natesan G, Chokkalingam A (2019) Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm. ICT Express 5:110–114. https://doi.org/10.1016/j.icte.2018.07.002
28. Hariharan B (2019) WBAT Job Scheduler: A Multi-Objective Approach for Job Scheduling Problem on Cloud Computing. J Circuits Syst Comput. 29. https://doi.org/10.1142/S0218126620500899
29. Devi K, Deep Learning Based Security Model for Cloud based Task Scheduling. KSII Transactions on Internet and Information Systems, (2020), 14(9) https://doi.org/10.3837/tiis.2020.09.005
30. T Essam H, Ahmed G, Yaser M, Ponnuthurai N. Task Scheduling in Cloud Computing based on Meta-heuristics: Review, Taxonomy, Open Challenges, and Future Trends, Swarm and Evolutionary Computation, (2021) 62, https://doi.org/10.1016/j.swevo.2021.100841.
31. Wanneng S, Ken C, Neal N (2021) Research on strong agile response task scheduling optimization enhancement with optimal resource usage in green cloud computing. Futur Gener Comput Syst 124:12–20. https://doi.org/10.1016/j.future.2021.05.012
32. Panda SK (2022) Shradha Surachita Nanda, Sourav Kumar Bhoi, A pair-based task scheduling algorithm for cloud computing environment. J King Saud Univ ComputInform Sci 34(1):1434–1445. https://doi.org/10.1016/j.jksuci.2018.10.001
33. Jain DK, Boyapati P, Venkatesh J, Prakash M. (2022). An intelligent cognitive-inspired computing with big data analytics framework for sentiment analysis and classification. Inform Proc Manag. 59(1):102758. https://doi.org/10.1016/j.ipm.2021.102758
34. Ramalingam C, Mohan P (2021) Addressing semantics standards for cloud portability and interoperability in multi cloud environment. Symmetry. 13(2):317. https://doi.org/10.3390/sym13020317
35. Saravana Kumar C. (2017) An authentication technique for accessing de-duplicated data from private cloud using one time password. Int J Inform Sec Privacy (IJISP). 11 (2): 1–10. https://doi.org/10.4018/IJISP.2017040101.
36. Arul Vinayakam Rajasimman M, Manoharan RK, Aridoss M, Galety MG. Robust facial expression recognition using an evolutionary algorithm with a deep learning model. App Sci. 2023; 13(1):468. https://doi.org/10.3390/app13010468
37. Shanmugavadivel K, Sathishkumar VE et al (2022) Deep learning based sentiment analysis and offensive language identification on multilingual code-mixed data. Sci Rep 12:21557. https://doi.org/10.1038/s41598-022-26092-3
38. Parthiban S, Harshavardhan A, Neelakandan S, Prashanthi V, Alhassan Alolo AR, Velmurugan S. (2022) "Chaotic Salp Swarm Optimization-Based Energy-Aware VMP Technique for Cloud Data Centers", Comput Intell Neurosci. https://doi.org/10.1155/2022/4343476
39. Ezhumalai P, Prakash M (2022) A Deep Learning Modified Neural Network(DLMNN) based proficient sentiment analysis technique on Twitter data. J Exp Theor Artif Intell. https://doi.org/10.1080/0952813X.2022.2093405
40. Gangathimmappa M, Subramani N, Sambath V, Ramanujam RA, Sammeta N. Deep learning enabled cross-lingual search with metaheuristic web-based query optimization model for multi-document summarization. Concurrency Computation Practice Experience. 2022;e7476. doi:https://doi.org/10.1002/cpe.7476
41. Reshmy AK, Paulraj D. (2019) Data mining of unstructured big data in cloud computing International. J Bus Intel Data Mining. 13(3):147–162

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.