Soft Computing (2022) 26:13069–13079
https://doi.org/10.1007/s00500-021-06488-5

FOCUS

Task scheduling in cloud computing using hybrid optimization algorithm

Mohd Sha Alam Khan[1] · R. Santhosh[1]

Accepted: 25 October 2021 / Published online: 6 November 2021

Abstract
Cloud computing provides a wide variety of services, from small to big businesses, to individual consumers. Cloud computing's features entice users to migrate their operations from traditional platforms to cloud platforms. In comparison to traditional systems, cloud computing has an extremely powerful processing capacity. Requests for resources are considered tasks in the cloud, and appropriate resources are allocated depending on user needs. However, owing to high demand and volume of requests, cloud struggles to allocate resources. Task schedulers are employed in cloud computing to address these issues. Various task scheduling methods have been presented in several research publications, and the quest for a better scheduling model continues. In this paper, a task scheduling method based on a hybrid optimization algorithm is presented, which effectively schedules jobs with the least amount of waiting time. In addition to these, other parameters, such as the overall production time, execution time, waiting time, efficiency, and utilization are included in this study. The simulation results show that the proposed scheduling method is superior to conventional Ant Colony and Particle Swarm Optimization-based scheduling algorithms in terms of performance.

Keywords Cloud computing · Task scheduling · Optimization

1 Introduction

Cloud computing introduces a revolution in information technology and other sectors through its efficient and powerful architecture. Cloud becomes a predominant solution for complex computing and big scale data operations. The elasticity in cloud computing can support numerous applications at an instance of time using virtual machines. The multi-tenant computing environment shares the resources with the users. Based on the user requests, the resource status is checked and allocated through an efficient scheduler module. To achieve real-time optimal performance in cloud computing an efficient scheduler plays a vital role. Scheduling algorithms map the jobs into cloud environment and utilize the available resources, thereby it reduces the latency, response time for the request and increases the resource utilization and throughput of the system.

The tasks in the cloud computing environment differ due to their processing characteristics and requirements. Based on these factors the tasks are classified into independent and dependent jobs. The dependent jobs commonly termed as workflow have higher priority jobs. Based on the priority, dependent jobs are further categorized into non-preemptive and preemptive. The non-preemptive job does not change or interrupt the system, whereas preemptive tasks interrupt the system to process first by holding the current process. Once the priority task is completed the current tasks are resumed. Jobs with deadline constraints have higher priority to meet the SLA (Service level agreement) requirements. The task that does not depend on other tasks is referred as independent tasks. Figure 1 depicts an illustration for scheduling in cloud computing.

The main objective of task scheduling in cloud computing is to map the jobs to the respective virtual machines. It is important to consider task the factors in task

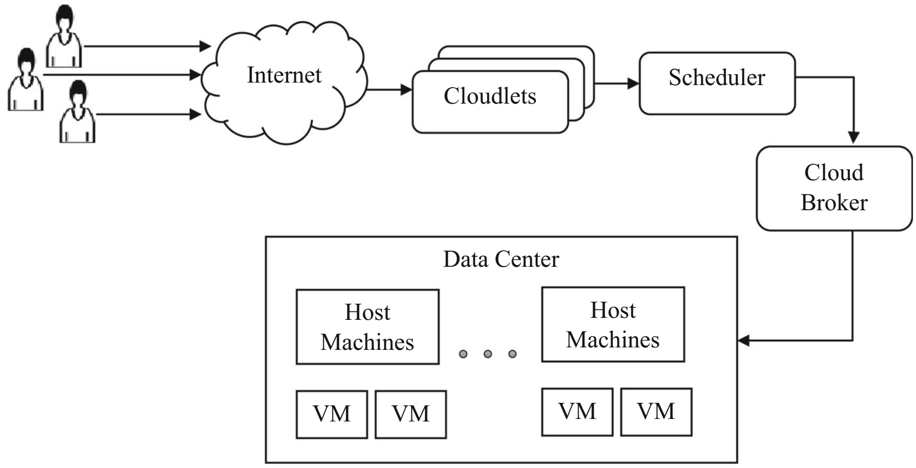
Communicated by Joy Iong-Zong Chen.

✉ Mohd Sha Alam Khan
mohdshaalamkhan12@gmail.com

R. Santhosh
santhoshrd@gmail.com

[1] Department of Computer Science and Engineering, Faculty of Engineering, Karpagam Academy of Higher Education, Coimbatore, India

Fig. 1 Scheduling in cloud computing



scheduling to utilize the resources so that the tasks can be completed with minimum time and increases the resource utilization of the cloud computing model. Similarly, load balancing can be achieved effectively through task scheduling. The scheduler distributes the workload so that all the resources can be utilized efficiently and avoids overutilization and underutilization. The role of cloud broker is to act as a negotiator between service provider and service applications to meet QoS requirements. On behalf of software services, cloud broker allocates resources to the application and manages the QoS standards. Therefore, allocating resources to desired tasks is challenging research in cloud computing. Generally, scheduling can be applied at the virtual machine level and application level (Kumar et al. 2019). The tasks are clustered and allocated to the suitable virtual machine based on the specification and other factors. So it is essential to consider all the parameters of virtual machines while designing a scheduler so that all the cloudlets and resources can be optimally utilized.

There are numerous metaheuristic approaches are introduced for efficient task scheduling and obtains optimal solutions, whereas heuristic approaches are problem dependent and it is not well suited for different datasets. So metaheuristic-based optimization models are widely preferred to overcome the scheduling issues. Over the years, numerous task scheduling algorithms are introduced such as Round Robin, Shortest Job First, First come First Serve, Max–Min, Multilevel queue, Min-Min (Arunarani et al. 2018) to attain better scheduling performance. However, these systems lag in performance in terms of higher waiting time. So it is essential to introduce an optimal system to reduce the waiting time which is the major objective of the research work.

The major contributions of the research are given as follows:

- Proposed a task scheduling model using a hybrid Particle Swarm Grey Wolf Optimization (PSGWO) algorithm for a cloud computing environment.
- To meet the QoS requirements, task execution time and virtual machine status are considered.
- Conducted experiments by varying the tasks using the NetBeans tool and evaluated the performance of the proposed scheduling algorithm in terms of makespan, execution time, waiting time, efficiency and utilization.
- Comparative analysis of proposed hybrid Particle Swarm Grey Wolf Optimization (PSGWO) and existing Ant colony optimization and particle swarm optimization based scheduling is presented and discussed in detail.

The rest of the article is structured as follows: a vast literature analysis to observe the merits and demerits of existing literature works are presented in Sect. 2, proposed hybrid task scheduling model is presented in Sect. 3, the performance of proposed scheduling algorithm is evaluated and the results are presented in Sect. 4 and finally, the observations and features of proposed work are concluded in Sect. 5.

2 Related works

Task scheduling is an important factor in cloud computing so that resources can be efficiently utilized and the requirements of the users can also be fulfilled. Various task scheduling algorithms are introduced by researchers in the past years and some of the relevant works are discussed in this section. Trust aware task scheduling model reported in (Rjoub et al. 2019) improves the scheduling performance and also improves the security features of cloud computing. The three state process computes the virtual machine trust level, prioritizes the tasks, and finally schedules the task as trust aware approach. The trust model attains better performance with minimized makespan compared to existing techniques. Task scheduling length is considered to avoid SLA violations and deadline constraints in workflow applications. It is crucial to minimize the task scheduling length for parallel tasks in cloud systems. To attain this, an efficient priority and relative distance algorithm is presented in (Muguntha 2020) that prioritizes the tasks in the first phase and maps the tasks to virtual machines based on the relative distance thus increases the resource utilization of the cloud.

Energy efficient dynamic offloading is reported in (Guo et al. 2019) along with resource scheduling to reduce energy consumption and task completion time. The task dependency requirements and deadline constraints are considered as energy efficient factors to reduce the completion time in the scheduling process. Heterogeneous earliest finish time techniques for task scheduling is reported in (Liu et al. 2018) provides an ideal solution based on the similarity of resource requests. Using non-dominated sorting the optimal solutions are obtained and the weight value for the requests is adjusted through an adaptive weight adjustment strategy that increases the scheduling performance in cost and deadline constraint meeting rate. Task scheduling in hybrid clouds reported in (Raj 2020; Li et al. 2018) considers the bandwidth in inter-cloud links to reduce the makespan and computation cost while executing an application. The information about available bandwidth is provided as input to the scheduler and resources are allocated. This increase the possible number of solutions for the given task to the scheduler, so that the scheduler can obtain efficient scheduling decisions.

Neural network based task classification and scheduling models are developed to improve the resource utilization of cloud computing. The intensive neural network layers compute the task requirements and virtual machine status and allocate resources to the users. Deep neural network based applications have high computational abilities and it is difficult to run directly with limited resources. So to reduce the complexities a feasible offline computation is utilized in few research work and it increases delay and affects the user performance. To overcome this, greedy (Chen et al. 2020a) and genetic algorithm based methods are reported in (Cui et al. 2020) that schedule the task for neural network applications based on the optimal solutions. However, the execution time of the genetic algorithm is high which is the major limitation of the research model.

Scheduling based on reinforcement learning reported in (Ding et al. 2020) minimizes the makespan and average waiting time considering the resource availability and deadline constraints. The learning process reported in the research work improves the convergence performance of Q-Learning model and balances the resource request for multiple tasks. Q-learning based task scheduling reported in (Tong et al. 2020) considers the energy efficiency of the resources and user requirements and then assigns the tasks to the respective machines. The two phase scheduling approach includes a centralized task dispatcher in the first phase that uses queuing model to assign the request to the cloud server. The Q-learning based scheduler in the second phase prioritizes the task based on lifetime, tolerance and assign task to the virtual machines. Similar Q-learning model reported in (Sivaganesan 2021) reduces the issues while handling directed acyclic graph (Aloboud and Kurdi 2019) in task mapping process as Deep Q-Learning task scheduling. The feature benefits of deep neural networks and Q-learning algorithm are combined to improve the makespan and load balancing characteristics of cloud task scheduling process.

Various optimization algorithms are used to obtain optimal solution in task scheduling process. The Cuckoo search optimization based task scheduling algorithm reported in (Peng et al. 2019) identifies optimal resources for the task without considering priority, because the priority based model prioritises high priority tasks first and low priority tasks must wait for a long time to access the resources. This optimization based scheduling gives equal priority for all the tasks and allocates optimal resources by analyzing the user requirements. However, due to this some of the resources are over utilized and others are underutilized that lead into issues in load balancing. Another optimal task scheduling reported in (Jia et al. 2021) utilizes whale optimization algorithm and allocates the resources to the tasks by considering the execution position, sequence, operating voltage and frequency. Optimization model provides better tradeoff between energy consumption and completion time through its feasible solutions and attains better performance compared to traditional scheduling models.

Adaptive ant colony optimization algorithm reported in (Sungheetha and Sharma 2021) meets the deadline constraints tasks based on self-adaptive weight factor and reduces the task execution time based on virtual machine characteristics. Fuzzy modified particle swarm optimization based task scheduling reported in (Chen et al. 2020b) improves the throughput and load balancing characteristics. The global search capability of the hybrid model is increased through roulette wheel selection, and velocity updating methods are modified using fuzzy PSO approach. Finally, the fitness function is obtained through fuzzy inference system. Hybrid model attains better performance in terms of makespan, efficiency and execution time compared to other techniques. Multi-objective optimization model for task scheduling reported in (Domanal et al. 2020) applies whale optimization algorithm to improve the scheduling and resource utilization performance of cloud computing. The improved model increases the searching ability of the optimization approach so that the scheduling model identifies optimal resources with minimum time and improved accuracy.

Hybrid optimization algorithm reported in (AbdElaziz et al. 2019) allocates the task to virtual machines based on the task demands, resource availability. Modified version of particle swarm optimization and cuckoo search optimization algorithm is incorporated and obtained a hybrid model. Improved reliability, better utilization of resources and minimized response time are the merits of the hybrid optimization approach. Increased computation cost is the observed demerit of the research work. Moth search algorithm (MSA) using differential evolution (DE) reported in (Yi Gu 2020) schedules the tasks to different VMs with minimum makespan. The exploration ability of the moth search algorithm is improved through differential algorithm. Bat optimization algorithm based task scheduling reported in (Dhaya and Kanthavel 2021) minimizes the computation time, energy consumption and provides optimal resource utilization. Other than the above discussed algorithms, few other optimization algorithms such as resource constrained task scheduling profit optimization (RCTSPO) (Hosseinioun et al. 2020), Hybrid Invasive Weed Optimization and culture evolutionary algorithm (Sahni and Vidyarthi 2018) are evolved for task scheduling. The scheduling process in this optimization models includes task clustering, classification and task scheduling for resource constrained task requests.

Cost effective scheduling algorithm reported in (Zhang and Zhou 2018) schedules the deadline constrained tasks considering the virtual machine performance ability and delay factors. Bayes classifier (Chen et al. 2018 ; Manoharan 2021) design for task classification model utilizes historical scheduling data to reduce the virtual machine creating time for similar resource requests in future. However, it requires large computational memory and identifies similar tasks and this increases the computation cost of the system. Bipartite Graph modeling based task scheduler reported in Manoharan (2021) obtains optimal solution for deadline constrained tasks through its graph model. Dynamic execution time and node performance are considered in the research work and allocate resources to the deadline constrained tasks by shortening the data access time and reduces the number of deadline constrained jobs.

From the above survey, it is observed that particle swarm optimization-based scheduling is widely preferred. However, the performance of optimization model can be increased using hybrid models for better selection of optimal virtual machines. In case of cuckoo search optimization-based approaches, the performance reduces due to slow convergence rate and local optimal solution issue. In the case of ant colony optimization, the convergence time is uncertain and it requires dependent sequences for random decisions. Traditional scheduling approaches like fuzzy based, neural network-based approaches consider the task priority and do not check the virtual machine status which leads into poor resource utilization. Similarly, the waiting time of the traditional models should be increased to attain better quality of services. Considering these limitations as motivation, this research work proposed a hybrid optimization model for efficient task scheduling in cloud computing environment.

3 Proposed work

The proposed hybrid optimization model for efficient task scheduling is presented in this section. The virtual machine status is identified before the optimization process so that optimal virtual machines can be selected for resource requests. Hybrid optimization approach incorporates the particle swarm optimization and grey wolf optimization algorithm for optimal resource allocation. For VM classification simple support vector machine is used in the proposed work. Support vector machine is well known for data classification applications. The simple machine learning approach provides better solution for structured and unstructured data. SVM does not fall into local optima and performs better than neural network models. Due to this functional benefits SVM is used in this research work to classify the virtual machines. So that tasks can be scheduled depends on the virtual machine status.

Figure 2 depicts the sample outputs of support vector machine. Similarly, the virtual machine states are classified and categorized into four different classes. To achieve this, the nonlinear and linear parameters of virtual machines are considered so that the utilization of virtual machines can be identified. Average utilization is directed as linear parameter, whereas over or underutilization is directed as nonlinear parameter. The boundaries between the parameters are obtained as classification results of support vector

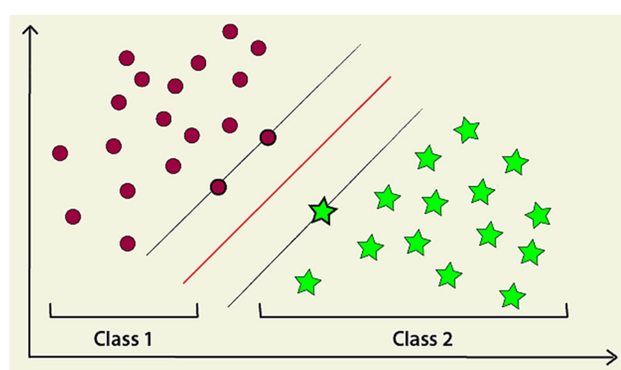


Fig. 2 SVM classifier

machine. Based on the results, the stability and utilization of the virtual machines can be identified. In the proposed work, the results of support vector machine categorize the virtual machine status into four classes such as stable, moderately stable and unstable. Based on this, the resource utilization is defined into underutilized, moderately utilized and overutilized.

The scheduling problem in the cloud computing is formulated into optimization problem considering multiple factors using hybrid optimization algorithm. The task priorities are assigned as weights and computed optimally through hybrid Particle Swarm Grey Wolf Optimization (PSGWO) model. The energy and load balancing indexes are assigned with higher weights and the overall weight function will be unity. The general optimization model is formulated into

$$f(x) = \sum_{n} w_n f(x_n)\ for\ 0 \leq n \leq j \tag{1}$$

where the weight function is represented as w_n, and the individual fitness function of optimization models are represented as $f(x_n)$ in the range $0 \leq n \leq j$. In order to attain optimal solution, the fitness function must be obtained as maximum value. Based on the fitness function, the task scheduling is formulated as

$$F(x) = e_t w_1 + c_c w_2 + \rho w_3 + c_t w_4 + r_t w_5 \tag{2}$$

where total energy required to complete the tasks is represented as e_t, the computation cost is represented as c_c, the load balancing factor is represented as ρ, the time taken to complete the task is represented as c_t and the response time is given as r_t. The weights are represented as w_1 to w_5 in the range of [0,1]. Consider a set of tasks $T = \{x_1, x_2, x_3 \ldots x_n\}$ and the virtual machines be $V = \{v_1, v_2, v_3 \ldots v_n\}$ The parameters that need to be computed based on execution time and processing time are task completion time, response time, energy, load balance, and cost. Mathematically the completion time is formulated based on the difference between the finishing time (f_t) and starting time of the task (s_t) for a virtual machine and it is given as

$$c_t = \sum_{t=1}^{n} f_{xt} - s_{xt} \tag{3}$$

The response time is formulated based on the difference between the submission time (sm_t) of the task and waiting time (w_t) of the task for a virtual machine and it is given as

$$r_t = \sum_{t=1}^{n} sm_{xt} - w_{xt} \tag{4}$$

The energy is estimated based on the depleted power level while executing the task considering the execution time and resource utilization and it is given as

$$e_t = p(u_{xt}) * c_t \tag{5}$$

where u_x is the utilization factor and it is expressed as $u_x = \frac{u_{ref} \times s_{ref}}{v_{ns}}$. The terms u_{ref} denotes the task resource consumption on n^{th} virtual machine, s_{ref} indicates the clock speed of reference virtual machine and v_{ns} indicates the virtual machine clock speed. The load balancing factor for different virtual machine process is computed based on the difference between the actual load (l_x) and average load $(\bar{l}_x)$ and it is given as

$$\rho = \sqrt{\frac{\sum_{v=1}^{n} l_{xt} - \bar{l}_{xt}}{n}} \tag{6}$$

Finally, the computation cost c_c is obtained based on the virtual machine processing capacity and task size and it is given as

$$c_c = \left(\frac{t_{x_n}}{p_{xn}}\right) \times c_n \tag{7}$$

Figure 3 depicts the simple process flow of proposed hybrid optimization model for task scheduling. In order to select the optimal virtual machines from the classifier results the hybrid optimization model incorporates particle swarm optimization and grey wolf optimization. The fitness function of particle swarm optimization is used to improve the fitness function of grey wolf optimization so that optimal solutions can be obtained for the given issue. PSO is well known for its simple implementation, high computational efficiency, robustness to control parameters and perform better than other heuristic optimization models. Before formulating the optimization model, few factors are assumed. The virtual machines are initially at minimum workload and all the virtual machines have sufficient energy to process the requests. For VM localization particle swarm optimization is used in the proposed approach. Consider different classes of virtual machines as $V = \{v_1, v_2, v_3 \ldots v_n\}$ and consider it as particles. The position and velocity of the particle swarm optimization are defined for virtual machines as $Vp = \{vp_{n1}, vp_{n2} \ldots vp_{ni}\}^T$ and $Vv = \{vv_{n1}, vv_{n2} \ldots vv_{ni}\}^T$. Where i is the space.

The maximum population for an individual obtained as best solution and it is given as

$$sbest_n = \{sbest_{v1}, sbest_{v2} \ldots sbest_{vn}\}^T \tag{8}$$

The global maximum is obtained as

$$gbest_n = \{gbest_{v1}, gbest_{v2} \ldots gbest_{vn}\}^T \tag{9}$$

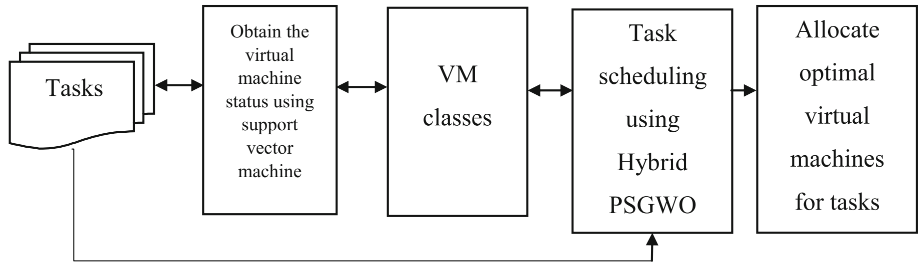


Fig. 3 Proposed hybrid optimized task scheduling process

Based on the velocity the particle positions are updated and it is given as

$$Vv_i^{z+1} = I_c vv_n^k + \varphi_1 rand(0,1)\left(sbest_i^k - Vp_n^k\right) + \varphi_2 rand(0,1)\left(gbest^k - Vp_n^k\right) \tag{10}$$

where the constants φ_1 and φ_2 represent the acceleration factor, and the inertia coefficient is given as I_c. To improve the optimal virtual machine search process, the coefficient functions are adjusted in the search process and the updated velocity and position is given as

$$Vv_n^{k+1} = \frac{1}{2}\left(sbest_n^k + sbest_{rand()}^k\right) \tag{11}$$

$$Vp_n^{k+1} = \frac{1}{2}\left(Vp_n^k + Vp_{rand()}^k\right) \tag{12}$$

To improve the optimality of the task scheduling process it is essential to reduce the task waiting time and to improve the task response time optimal virtual machine must be allocated with minimum waiting time. In order to enhance the optimum selection of virtual machine for the corresponding resource request, Grey wolf optimization (GWO) is included in the proposed work. Compared to other optimization algorithms, GWO provides fast convergence, high precision and it is well known for simple shortest path optimization problems. The hunting nature of grey wolves is formulated as an optimization model such as tracking, encircling and attacking the prey. The optimal virtual machines selected through particle swarm optimization are further processed through the grey wolf optimization. This process reduces the task waiting time as the grey wolf optimization model assigns the resources immediately once the VM is identified to be suitable for the given task. The optimization model first selects the optimal virtual machine and analyzes the characteristics based on user requests and once it is confirmed as optimal VM then it is allocated. This process is directly related to attacking characteristics in which the optimal virtual machine is the prey and the tasks are the wolves so that the task with high computational demand for the resource request is identified as alpha and next category is grouped as beta and the remaining are considered as delta. The solutions are formulated and it is given as

$$\begin{aligned} D_\alpha &= \left|v_1 \times s_\alpha - S(t)\right| \\ D_\beta &= \left|v_2 \times s_\beta - S(t)\right| \\ D_\delta &= \left|v_{n-2} \times s_\delta - S(t)\right| \end{aligned} \tag{13}$$

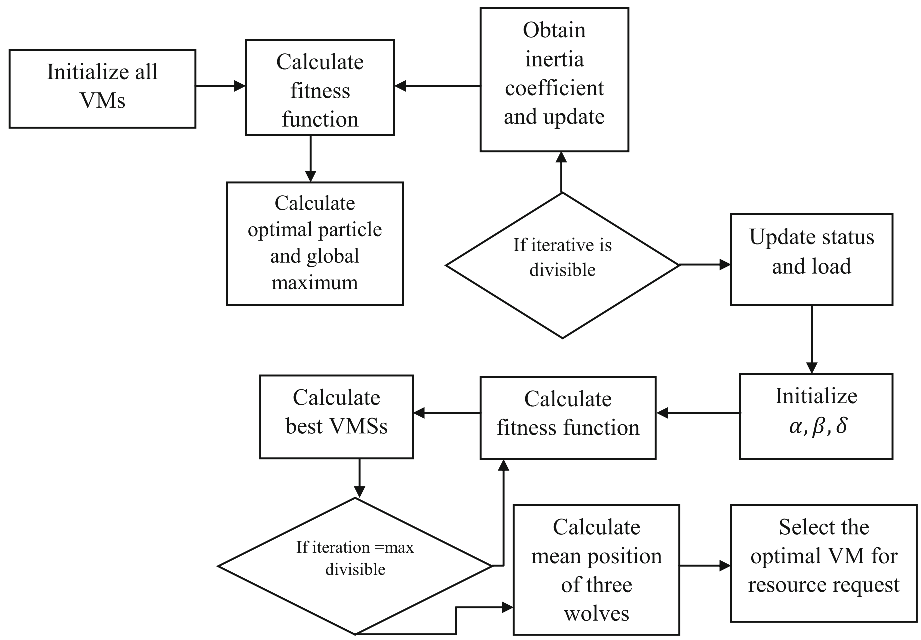
The position of prey is expressed as $S_P(t+1)$ and based on Eq. 11, it is obtained as

$$S_P(t+1) = \frac{S_1 + S_2 + S_3}{3} \tag{14}$$

where$S_1 = |s_\alpha - l_1 D_\alpha|$, $S_2 = \left|s_\beta - l_2 D_\beta\right|$ and $S_3 = |s_\delta - l_2 D_\delta|$ that represent the high computation resource requirement tasks. In the search process finding optimal solution is defined based on $\mathcal{L}$ which is the range {-2$\mathcal{L}$, 2 $\mathcal{L}$} and if the value is greater or equal to one then the process is stopped and next virtual machine is selected as optimal solution. The process of proposed hybrid optimization model for task scheduling is given in Fig. 4.

The process starts from initializing the particles which is related to virtual machines and then fitness function of each particle is calculated. Then Optimal particle and global maximum are obtained based on the fitness function, and the positions are updated to the new position if the iterative is divisible otherwise the process is repeated by obtaining the inertia coefficient. Once the best virtual machines are obtained those optimal virtual machines are further utilized by grey wolf optimization to select appropriate virtual machine with minimum waiting time. The process starts by evaluating the fitness function for selected optimal virtual machines and it is calculated based on the energy, load balancing and execution time. From that, the best wolves are selected as optimal virtual machines once maximum iteration is attained otherwise the process is repeated till the selection of optimal virtual machine for the given resource request. The proposed model initially utilizes machine learning models for virtual machine status identification. Based on the results, optimal virtual machine is allocated for the given tasks. Incorporating machine learning and hybrid optimization in task scheduling is the novelty of this research work. Earlier single optimization models are employed for task scheduling, but in the

Fig. 4 Process flow of proposed hybrid optimization model



proposed work hybrid optimization includes two optimization techniques for better improvement of results.

The pseudocode for the hybrid optimization task scheduling in cloud computing is summarized as follows:

Algorithm: Optimal VM selection using Hybrid PSGWO model

Input: Particles $s_z^{(i)}$ *for i = 1,2…n,* $\alpha, \beta, \delta, \vartheta, v$

Initialize: $T = \{x_1, x_2, x_3 \ldots x_n\}$, $V = \{v_1, v_2, v_3 \ldots v_n\}$, *weights* w_n, w, g, $sv_z^{(i)}$,

Calculate the fitness function for each VM

Calculate optimal particle from fitness value and obtain global maximum

update local and global maximum based on the fitness value of current particle

Check the iteration (z) is less than max generation

if z < max generation

Increase iteration by 1 and repeat

else Calculate the fitness function

Calculate the best wolves

search: If iteration reached max

Calculate the mean position of three wolves

ElseUpdate the positions

Repeat search

end

Table 1 Simulation parameters

S. No.	Parameter	Value
1	Number of data center	10
2	Number of tasks	500–1000
3	Number of Hosts	20
4	VM used	100
5	Memory capacity	100 GB

4 Results and discussion

The proposed hybrid optimized task scheduling model performance is evaluated through simulation using NetBeans 8.1. The operating system is Windows 10 installed on i3 processor of 2.20 GHz frequency. The proposed model performance is evaluated using the parameters such as makespan, average waiting time, average execution time, fitness function, energy, resource utilization and efficiency. Conventional Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO) based scheduling models are compared with proposed hybrid PSGWO model. The simulation parameters used for the proposed scheduling model are listed in Table 1, and it describes the number of virtual machines, data centers, host machines, number of tasks and memory.

Figure 5 depicts the comparative analysis of makespan for 1000 tasks. The tasks count is increased by 100 and the performance is observed. Compared to proposed hybrid optimization based task scheduling the performance of

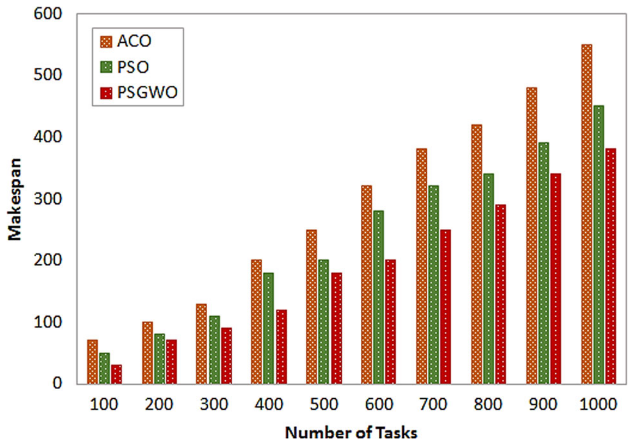


Fig. 5 Makespan comparison

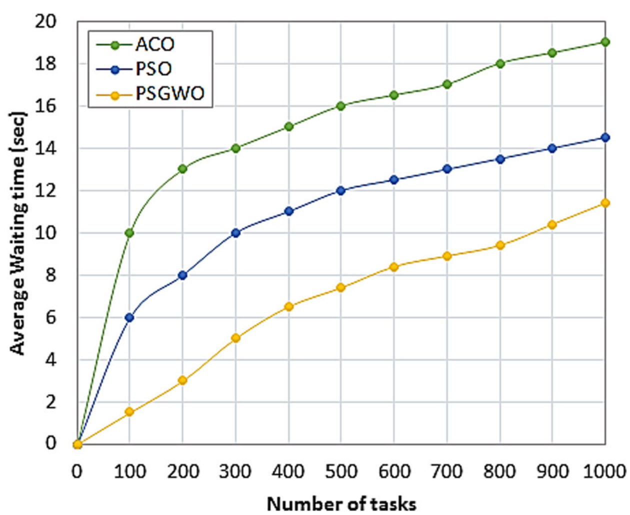


Fig. 6 Average waiting time

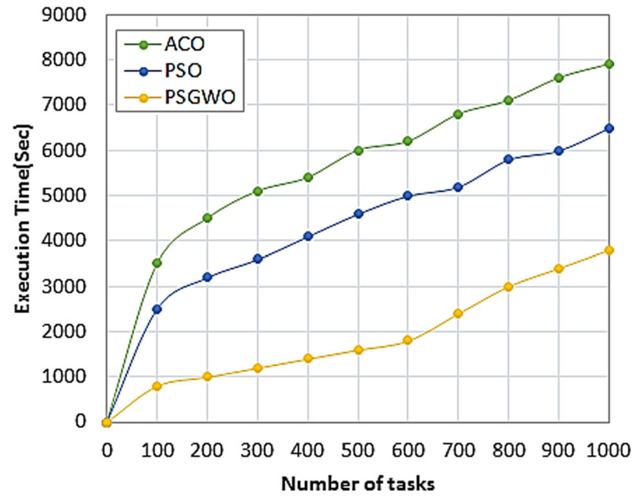


Fig. 7 Average execution time

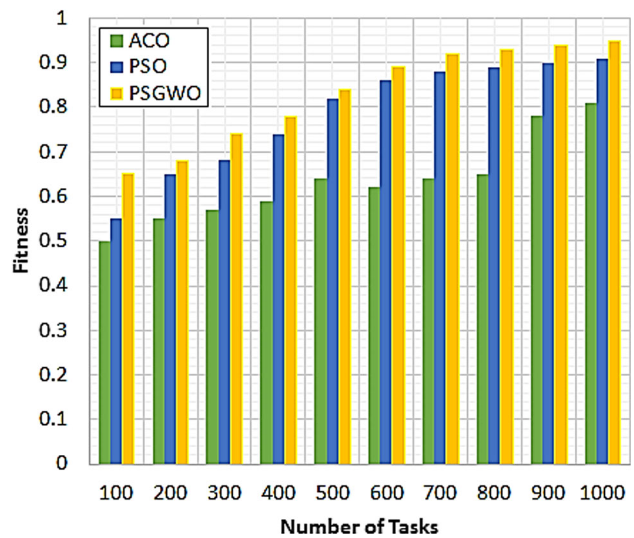


Fig. 8 Fitness calculation

existing methods are less due to the absence of virtual machine classification process, whereas the classification results clearly provide the virtual machine status so that selecting optimal virtual machine from the results is simple in the proposed approach. Due to this benefit, the makespan of the proposed approach is minimized even for maximum task counts.

The average waiting time and execution time for the proposed hybrid optimization model are compared with existing optimization models and depicted in Fig. 6 and Fig. 7, respectively. It is observed from the analysis, the proposed model exhibits minimum waiting time and execution time for the tasks compared to existing optimization models. Due to minimum waiting time, the response time of the proposed model is also reduced and satisfies the necessary quality of services requirements.

The fitness for the optimization model is compared in Fig. 8 with respect to total number of tasks. The maximum fitness value indicates the optimal solution of the scheduling model. Proposed hybrid optimization model has maximum fitness values compared to existing techniques. Two fitness functions are considered in the proposed approach and the first fitness is used to improve the fitness parameter of grey wolf optimization model and obtains optimal solution.

The energy analysis for the proposed approach and existing approaches is depicted in Fig. 9. It is observed from the figure that the proposed model has minimum energy requirements to process the requests and for maximum VM ratio the requirement is less compared to other techniques. Initially the energy requirements are low for all the models. Specifically proposed scheduling approach has minimum energy requirements compared to PSO and ACO model. The ratio is further increased to maximum, and the

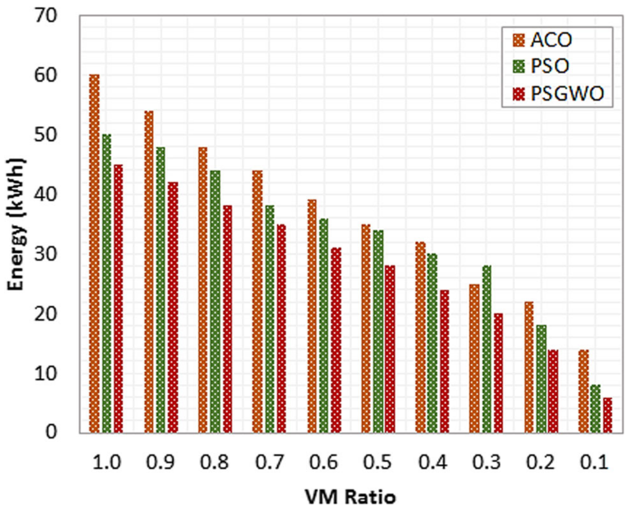


Fig. 9 Energy analysis

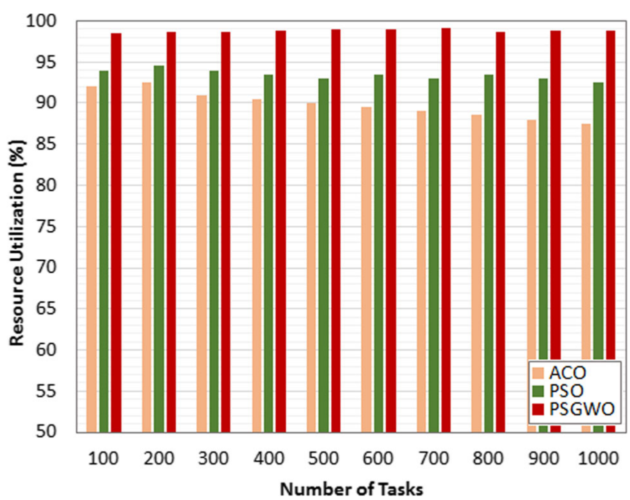


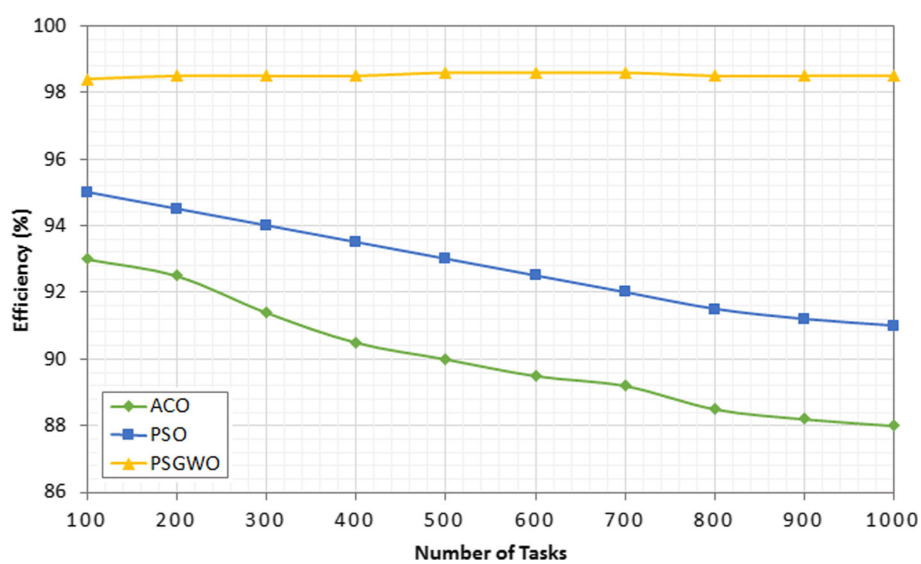
Fig. 10 Resource utilization

results are observed. For maximum VM ratio, the energy requirement of proposed approach is 45kWh, whereas PSO attains 50kWh and ACO attains 60kWh.

Figure 10 depicts the resource utilization analysis of proposed optimization model and existing optimization model as a comparative analysis. It is observed from the results that the proposed hybrid optimization model has maximum resource utilization by selecting optimal virtual machines for the resource requests. The performance of conventional particle swarm optimization is quite better, and ant colony optimization has less resource utilization compared to proposed hybrid optimization model. The resource utilization of PSO model is better than ACO model; however, it is less than the proposed approach. The traditional PSO based scheduling is better in task scheduling but the fitness function is used to assign the task in traditional approach. But in the proposed work, the fitness function is used as parameter for GWO model so that optimal VMs are selected in GWO model using the fitness function obtained from PSO.

The overall efficiency of the proposed model and existing models are compared based on the number of tasks and depicted in Fig. 11. Efficiency is observed based on the resource utilization, waiting time and execution time. The system must provide better services with maximum utilization and minimum execution time. Also, it is essential to consider the waiting time, if the waiting time is high, the chances of deadline miss may occur. So, tasks must be scheduled with minimum waiting time. Considering all these parameters the efficiency is measured for the proposed task scheduling model. It can be observed from that the figure the efficiency of the ant colony optimization and particle swarm optimization models reduces when the task reaches the maximum count, whereas proposed model attains stable efficiency from minimum number of tasks to

Fig. 11 Efficiency comparison



maximum number of tasks due to optimal selection of resources for the requests.

5 Conclusion

This research work presents a hybrid optimization approach for task scheduling in cloud computing to minimize the waiting time and to improve the quality of services. The virtual machines status is initially classified based on the loads using support vector machine. The hybrid approach using particle swarm grey wolf optimization (PSGWO) identifies optimal virtual machines and allocates the resources to the tasks. Proposed scheduling model is experimentally verified and compared with conventional ant colony optimization and particle swarm optimization based scheduling algorithms. Compared to existing techniques proposed hybrid optimization based task scheduling attains better performance in all parameters. Further, this research work can be extended toward virtual machine placement for QoS improvements.

Funding No funding.

Data availability We used own data and we used own coding.

Declarations

Conflict of interest We have no conflicts of interest to disclose.

Humans and animals participants Humans and animals are not involved in this research work.

References

AbdElaziz M, ShengwuXiong LL (2019) Task scheduling in cloud computing based on hybrid moth search algorithm and differential evolution. Knowl Based Syst 169:39–52

Aloboud E, Kurdi H (2019) Cuckoo-inspired job scheduling algorithm for cloud computing. Proc Comput Sci 151:1078−1083

Arunarani AR, Manjula D, Sugumaran V (2018) Task scheduling techniques in cloud computing: a literature survey. Futur Gener Comput Syst 91:407–415

Chen X, Cheng L, Liu C, Liu Q, Liu J, Mao Y, Murphy J (2020b) A WOA-based optimization approach for task scheduling in cloud computing systems. IEEE Syst J 14(3):3117–3128

Chen Z, Junqin Hu, Chen X, Jia Hu, Zheng X, Min G (2020a) Computation offloading and task scheduling for dnn-based applications in cloud-edge computing. IEEE Access 8:115537–115547

Chen C-H, Lin J-W, Kuo S-Y (2018) MapReduce scheduling for deadline-constrained jobs in heterogeneous cloud computing systems. IEEE Trans Cloud Comput 6(1):127–140

Cui D, Peng Z, JianbinXiong BX, Lin W (2020) A reinforcement learning-based mixed job scheduler scheme for grid or IaaS cloud. IEEE Trans Cloud Comput 8(4):1030–1039

Dhaya R, Kanthavel R (2021) Bus-based VANET using ACO multipath routing algorithm. J Trends Comput Sci Smart Technol (TCSST) 3(01):40–48

Ding D, Fan X, Zeng J (2020) Q-learning based dynamic task scheduling for energy-efficient cloud computing. Futur Gener Comput Syst 108:361–371

Domanal SG, Guddeti RMR, Buyya R (2020) A hybrid bio-inspired algorithm for scheduling and resource management in cloud environment. IEEE Trans Serv Comput 13(1):3–15

Gaith Rjoub JB, Wahab OA (2019) BigTrustScheduling: trust-aware big data task scheduling approach in cloud computing environments. Futur Gener Comput Syst 110:1079–1097

Jia Y-H, Chen W-N, Yuan H, Tianlong Gu, Zhang H, Gao Y, Zhang J (2021) An intelligent cloud workflow scheduling system with time estimation and adaptive ant colony optimization. IEEE Trans Syst Man Cyber Syst 51(1):634–649

Jyoti Sahni DPV (2018) A cost-effective deadline-constrained dynamic scheduling algorithm for scientific workflows in a cloud environment. IEEE Trans Cloud Comput 6(1):2–18

Li X,Qian L, RuizR (2018) Cloud workflow scheduling with deadlines and time slot availability. IEEE Trans Serv Comput 11(2):329−340

Liu L, Fan Q, Buyya R (2018) A deadline-constrained multi-objective task scheduling algorithm in mobile cloud environments. IEEE Access 6:52982–52996

Manoharan JS (2021) A novel user layer cloud security model based on chaotic arnold transformation using fingerprint biometric traits. J Innov Image Process (JIIP) 3(01):36–51

Mohit Kumar M, SharmaSingh SCSP (2019) A comprehensive survey for scheduling techniques in cloud computing. J Netw Comput Appl 143:1–33

Mugunthan SR (2020) Novel cluster rotating and routing strategy for software defined wireless sensor networks. Journal of ISMAC 2(02):140–146

PejmanHosseinioun MK, Ghaemi R (2020) A new energy-aware tasks scheduling approach in fog computing using hybrid metaheuristic algorithm. J Parall Distrib Comput 143:88–96

Peng H, Wen W-S, Li L-L (2019) Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment. Appl Soft Comput 80:534–545

Raj JS (2020) Machine learning based resourceful clustering with load optimization for wireless sensor networks. J Ubiquit Comput Commun Technol (UCCT) 2(01):29−38

Sivaganesan D (2021) A Data Driven Trust Mechanism Based On Blockchain in IoT sensor networks for detection and mitigation of attacks. J Trends Comput Sci Smart Technol (TCSST) 3(01):59–69

Songtao Guo JL, Yang Y, Xiao B, Li Z (2019) Energy-efficient dynamic computation offloading and cooperative task scheduling in mobile cloud computing. IEEE Trans Mob Comput 18(2):319–333

Sungheetha A, Sharma R (2021) fuzzy chaos whale optimization and BAT integrated algorithm for parameter estimation in sewage treatment. J Soft Comput Paradigm (JSCP) 3(01):10–18

Tong Z, Chen H, Li K (2020) A scheduling scheme in the cloud computing environment using deep Q-learning. Inf Sci 512:1170–1191

Yi Gu C (2020) Energy-aware workflow scheduling and optimization in clouds using bat algorithm. Future Generat Comput Syst 113:106−112

Zhang PeiYun, Zhou MengChu (2018) Dynamic cloud task scheduling based on a two-stage strategy. IEEE Trans Autom Sci Eng 15(2):772–783