International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Hybrid Optimization Approach in Cloud Computing Based on Yellow Saddle Goatfish and Particle Swarm Optimization Algorithms for Task Scheduling

[1]Ms. Manpreet Kaur, [2]Dr. Sarpreet Singh

Submitted: 17/01/2024 **Revised**: 25/02/2024 **Accepted**: 03/03/2024

Abstract: Task scheduling is a necessary component of any distributed infrastructure since it distributes jobs to the appropriate resources, for the process of execution. The task scheduling algorithm presented in this research uses an integrated optimization approach for scheduling tasks seamlessly and effectively on cloud computing. In the proposed work, we have used Yellow saddle Goat Fish algorithm (YSGA) along with particle swarm optimization (PSO) algorithm. Initially, a random population is generated upon which hybrid YPSO model is implemented to attain fitness values. Here, the proposed hybrid YPSO model analyzes six factors i.e., cost, average completion time, make span time, consumption of energy during process, utilization of available resource and handling of load to calculate its fitness value. The iteration with the least fitness value will be selected as the final one and all the task will be schedules as per this fitness value. The performance of YPSO model is then analyzed and compared with standard YSGA model in MATLAB Software under two scenarios. In the first case, we analyzed performance of proposed model with respect to standard YSGA model for varying tasks with 3 VMs, while as, in second case VMs are varied. Simulating outcomes depict that in both cases the fitness value keeps getting better in proposed hybrid YPSO model to prove its supremacy over other similar methods.

Keywords*: cloud computing, load balancing, workflow scheduling, resource management, optimization algorithms.*

1. Introduction

Cloud computing has revolutionized the world of Information Technology in recent years. This cutting-edge technology offers a wide range of resources, including hardware, software applications, development platforms, and testing tools, all readily available through services. The platform or tools those are used for development and testing purposes comes under the category of Platform and Software as a Service, whereas the components those are used to develop the hardware are classified under Infrastructure as service i.e. IaaS. Cloud computing offers several benefits to consumers, including on-demand access to resources, scalability, flexibility, and virtualization-type services. To take full advantage of these benefits, it is essential to have an effective management system in place that can effectively utilize these resources. Resource management in cloud computing can be accomplished through the implementation of effective resource scheduling, allocation, and scalability approaches. The main goal of resource management is to use resources as efficiently as possible to meet the changing demands of the users under cloud environment. This can be achieved through the use of effective scheduling algorithms that allocate resources to tasks in an optimized manner and through the use of scalability approaches that allow the cloud environment to dynamically expand or contract based on the needs of the users. In conclusion, the effective management of resources in cloud computing is crucial to achieving the benefits of this innovative technology. By implementing and maintaining effective resource scheduling, allocation, and scalability approaches, cloud computing can continue to evolve and provide users with a cost-effective, scalable, and flexible solution for their computing needs.

A key element of the cloud's computing system is load balancing that aids in the equitable distribution of workloads among the resources that are on hand. In order to optimize efficiency, decrease delay, and maximize utilization of resources, load balancing aims to prevent any resource from becoming overloaded. In cloud computing environments, load balancing is particularly important because it helps to address several challenges that arise due to unpredictable user demand, resource constraints, and network latency [1].The significance of load balancing in cloud based computing cannot be emphasized enough. Effective load balancing helps to optimize the usage of available resources, leading to improved performance and reduced downtime. Load balancing also helps to prevent the overuse of resources, resulting in reduced costs and improved resource utilization [2]. By distributing workloads evenly across

[1]*Department of Computer Science, Sri Guru Granth Sahib World University,Fatehgarh Sahib (140406), Punjab, India.*
Email: manpreetkaur.mgc@gmail.com
[2]*Department of Computer Science, Sri Guru Granth Sahib World University,Fatehgarh Sahib (140406), Punjab, India*
Email: ersarpreetvirk@gmail.com

resources, load balancing ensures that cloud computing services are able to meet the growing demand for computing resources and deliver high-quality, reliable, and cost-effective solutions.

Another crucial element of computing in the cloud is the scheduling of tasks. It plays a very crucial role in balancing the load because it controls the sequence in which tasks are carried out on available resources. To achieve better load balancing in cloud computing, task scheduling must focus on optimizing resource utilization and ensuring that resources are used in a balanced manner. Task scheduling algorithms should take into account several factors such as resource constraints, task dependencies, and task priority, and they should dynamically adjust the execution order of tasks based on the current load on each resource [3].In conclusion, load balancing and task scheduling are interrelated components of cloud computing that work together to ensure the efficient and effective use of resources. By utilizing effective load balancing algorithms and task scheduling approaches, cloud computing services can offer users a more reliable and efficient experience, even during periods of high demand. Furthermore, it ensures that resources are used optimally, leading to reduced costs and improved resource utilization.

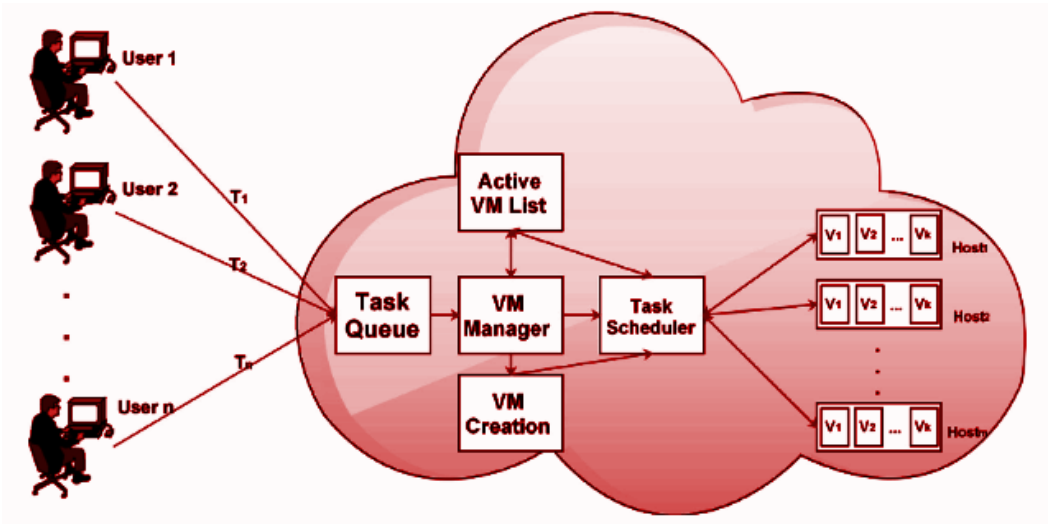

Fig.1: Model for Scheduling in a Cloud Data Center.

Several algorithms and techniques have been proposed for task scheduling in cloud computing, but it remains an important area of research due to the dynamic and heterogeneous nature of cloud environments. New and improved algorithms are needed to address the unique challenges posed by cloud computing and to meet the increasing demands for computational resources. This study intends to advance the area by investigating novel task scheduling strategies for computing in cloud environments.

The primary goal of the research is to lessen the shortcomings of the existing load balancing algorithms, which use a monotonic method of using one or two parameters and replace them with an algorithm that combines a variety of parameters for the load distribution and achieves the best distribution, which will be resource-efficient, fast, and accurate. A list of this paper's contributions is provided below.: Examine the elements that contribute to the issue of unbalanced load in cloud computing, Summarizing the load-balancing methods in detail that are currently in use and how they have been applied to cloud computing, Talk about a novel load-distribution method that takes numerous factors into account, Examine the difficulties researchers experienced when creating a successful load-balancing algorithm.

2. Literature Survey

This section addresses several task scheduling approaches as well as their advantages and disadvantages. The paper also explains the basic idea of the suggested scheduling strategy after looking at the existing approaches.

The author in paper [3] proposed a scheme that was working on providing a better solution for the resource allocation in the VMs. In order to achieve this the scheme was utilizing first come first serve (FCFS) methodology of load balancing while utilizing the nearest data center's broker policy. The authors also explained that the proposed schemes are contrasted with those of other well-known algorithms, such as the round robin(RR) and throttled algorithms. When analyzed further, it was found that in certain cases time for response was very low with traditional algorithms,

therefore main focus of the few researches were to improve the time of response. Tarandeep and Kriti Bhushan [5] were also focusing on improving the performance factors such as makespan time, response time, etc. They incorporate the system's overall scheduling module that included process as well as CPU based scheduling. They employed the met heuristic method ACO (Ant Colony Optimization), which allocates processes in accordance with processor capacity and FCFS and RR were performing the task of CPU scheduling. A comparison was also conducting with the variants of combined scheduling. The results demonstrated that the scheme utilizing the ant colony optimization was performing outstanding in comparison with variants as FCFS-FCFS and FCFS-RR combined models. The given algorithm was achieving improved makespan time and better response time too that was justifying the improved load balancing in the system. Using a modified variant of PSO based task scheduling method supports the load balancing termed as LBMPSO, Arabinda Pradhan Sukant and KishoroBisoy [6] suggested a technique in this study to allocate workloads among the cloud resources those are available for use in a manner that it minimizes the completion time and increases resource consumption. By making sure that the tasks and resources that exist inside the computing facility are appropriately informed, this can be achieved. Utilizing the CloudSim simulator, they put their suggested scheduling strategy into practice. Simulation findings clearly demonstrate that the proposed scheduling algorithm outperforms other current strategies in terms of minimizing makespan and improving resource consumption. An additional work was proposed in [7] that aims to offer an effective cloud based service. It main working scenario was based on the load balancing approach in collaboration with the effective resource utilization. The methodology that it followed was based on decision based system. The proposed scheme was utilizing the properties of fuzzy logic system based on multi dimensionality oriented resource scheduling modeling.In addition to this the scheme was having an additional phase that utilized Multidimensional Queuing Load Optimization algorithm. This phase's primary effort was to select the user's request in a flexible manner. This assists the proposed model to achieve an improved utilization scheme for virtual machines followed by better load balancing.Later that, the scheme was focused on setting up a load balancing mechanism to minimize latency for every request type and proving prevention method for over or under usage of resources. Via simulations, the Cloudsim simulator's application to data centers hosted in the cloud evaluations were conducted and the findings show that the suggested approach outperforms it when measured in terms of average success rate, resource planning efficiency, and response time. When compared to the most recent works, simulation analysis demonstrates that the strategy increases resource scheduling efficiency by 7% and decreases reaction time by 35.5%. An effective and progressive a variant of MBA approach that was utilizing whale optimization algorithm for multi-objective planning and scheduling in cloud - based solutions was suggested by Manikandan et al. [8]. The provided method's multi-objective nature decreased the makespan by maximizing the use of resources. For a random double adaptive WOA, the idea put forward was approved. The author stated that the algorithm's performance can be improved if it will be collaborated with the mutation operator of one other algorithm termed as bee algorithm. After designing and simulation, the proposed scheme was examined by using the cloudsim platform. The results were analyzed and compared under certain criteria and performance factors including the time and costing. The suggested scheme shows its effectiveness in terms of computational cost, completion and execution time. Even it was capable to utilize the resource in a much better way when compared with other available algorithms.In order to schedule tasks in a cloud environment, The Harris Hawks Optimization (HHO) method was presented in this paper [9] that relies on simulated annealing method termed as (SA). The typical HHO method's rate of convergence and solution quality are improved in given approach where role of SA phase was as a searching the local optimum solution. Utilizing the CloudSim toolkit, the HHO-SA method is compared to that of cutting-edge work scheduling methods. The effectiveness of the method is examined using both typical and simulated workloads. The collected findings show that, in comparison to normal HHO and other existing scheduling methods, HHOSA can significantly shorten the makespan of the work scheduling problem. In another research [10], a cloud-edge workflow scheduling problem is addressed using a modified Harris Hawks optimization technique. Cost and makespan are the two primary goals of simulations. The proposed experiments made use of actual workflow models to assess the proposed algorithm in comparison to other strategies examined in recent literature and put to the test in a similar simulation environment. The suggested improved Harris hawks optimization technique shows improvements in certain QoS factors when compared with existing approaches.

From the above given literatures, it is observed that over the last few years a number of approaches have been proposed by various scholars in order to increase the efficiency of cloud computing models. No doubt that these models were generating good results however; we observed that there is a scope of improvement. One of the primary drawbacks of the current work scheduling schemes was that they considered only few parameters

for performing the task scheduling effectively, however, there are number of other parameters that must be considered in order to make the system more efficient and effective. Moreover, majority of the authors have used different optimization algorithms in their works, but these optimization models either have slow convergence rate or tend to get trapped in local minima. In addition to this, we have also analyzed that very few researches have been done on using hybrid optimization algorithms. Considering these details, a novel and feasible task scheduling method must be put forward that can get over these constraints.

3. Proposed Work

In this work, a revolutionary and effective task scheduling model based on hybrid optimization techniques is developed to overcome the constraints of previous approaches. The main motive of proposed work is to schedule and optimize the tasks in cloud computing effectively so that overall performance of model is increased. To accomplish this objective, we have updated two important phases i.e. implementation of hybrid optimization algorithm and fitness value upgradation during task scheduling. In the proposed work, we have used new optimization algorithm i.e. Yellow Saddle Goat Fish Algorithm (YSGA) along with particle Swarm Optimization (PSO) algorithm. The main reason for using the given two optimization algorithms is that when looking for global solutions, they exhibit an elevated rate of convergence and avoid becoming stuck in local minima. Another reason for using the two algorithms i.e. YSGA and PSO together is to increase the efficiency of task scheduling by overcoming the limitations of each other. In addition to this, we have updated the performance of proposed model by updating the fitness value. Following analysis of the literature review, we have analyzed that it is important to consider all important parameters in the proposed work in order to achieve high level performance. In the proposed work, we have considered cost time, average completion time, make span time, energy consumption, resource utilization and load handling as six parameters that are analyzed for calculating the fitness value. The six fitness values are then grouped into three fitness factors i.e., ACET (Average completion and execution time including Make span and execution time), $\boldsymbol{E_c}$(Energy Consumption) and $R_u, LRHR$ (Resource utilization, Load resource handling ratio including load and VM capacity combined) in order to analyze the weights for each parameter for effective load scheduling. For every iteration, the best fitness value is stored and at the end, the least value of fitness will be selected as final and all the tasks will be scheduled based on this fitness. The different phases of proposed model are model initialization in which various parameters like total number of VMs, tasks, initial population, maximum iterations and values of YPSO constants are defined. After this random population is generated and hybrid optimization algorithms are implemented to calculate the fitness value. The detailed and step wise working of the proposed Hybrid YPSO based task scheduling model is explained in this section of paper.

a) Model Initialization

This refers to the initial stage of the suggested hybrid YPSO method, where each of the parameters are determined, including the overall count of VMs and tasks. Moreover, different YPSO parameters like population size, maximum iterations to be performed, no of clusters to be formed and value of other constants are defined. These parameters and constants values are specified and tabulated and are given in table 1.

Table 1: Simulation setup parameters

Simulation Setup	Values
No of VMs	3-25
No of Task	10,20,30,40,50
Population	10
Iteration	100
$\boldsymbol{\beta}$	[0 2]
α	1
ρ	[a 1]
a	[-1 -2] linearly decreases
b	1
λ	10
No of Cluster	4

The above tables represents that in the proposed work, we have used 3 to 25 VMs for 5 tasks i.e. 10, 20, upto 50. The initial count of population is 10 and iterations to be performed are 100. Moreover, a total of four clusters are formed in the network. These parameters complete the simulation set up for proposed model and next phases are implemented after this phase.

b) Population Generation

Once the model is initialized by defining various parameters, it is time for the next phase of proposed work wherein population is generated randomly. In the proposed work, population is generated randomly for n population by using the equation given in below equation.

For $\boldsymbol{Position_{ts}}$= $\{\boldsymbol{p}_1, \boldsymbol{p}_2,...,_t\}$, the value of random population is

$$\boldsymbol{Position_{ts} = randperm(n)} \quad \text{------- (1)}$$

Where, n is the population size.

c) Implement Hybrid YPSO for task scheduling

Next in the proposed model's development, Hybrid YPSO model is implemented on the randomly generated population in order to attain the best fitness value for task scheduling. The proposed Hybrid YPSO model analyzes six factors i.e. cost time, average completion time, make span time, energy consumption, resource utilization and load handling to calculate its fitness value. The fitness value is calculated by grouping the six factors into three factors as ACET (Average completion and execution time including Make span and execution time), $\boldsymbol{E_c}$(Energy Consumption) and $R_u, LRHR$ (Resource utilization, Load Resource Handling Ratio including load and VM capacity combined). By analyzing these factors, the value of fitness is calculated by using the below given equation.

$$\boldsymbol{fitness = w_1}\text{*ACET+}\boldsymbol{w_2*E_c+w_3*(R_u, LRHR)} \quad \text{-------- (2)}$$

Wherein, w1, w2 and w3 represent the weights of ACET, $\boldsymbol{E_c}$and $R_u, LRHR$ that depict the exact values of three factors in order to determine the best fitness. The best fitness in our case should be as least as possible and all tasks will be scheduled as per this fitness value. The Pseudo code for proposed Hybrid YPSO model is given in Algorithm 1.

Algorithm 1: Pseudo code for proposed hybrid YPSO model

1. Generating population n randomly for task scheduling $\boldsymbol{Position_{ts}}$=$\{\boldsymbol{p}_1,\boldsymbol{p}_2,...,\boldsymbol{p}_t\}$ using equation :

$$\boldsymbol{Position_{ts} = randperm(n)}$$

2. Calculatethefitness valueof every particle by using equation :

$$\boldsymbol{fitness = w_1}\text{*ACET+}\boldsymbol{w_2*E_c+w_3*(R_u, LRHR)}$$

3. Selectionofglobalbest fitness $\boldsymbol{g_{bfitness}}$
4. Separation of population$\boldsymbol{Position_{ts}}$into$\boldsymbol{k}$clusters$\{\mathbf{c_1,c_2,...,c_k}\}$
5. Selection of thechaser$\boldsymbol{Chaser_{position}}$andtheblocker$\boldsymbol{Blocker_{position}}$fishforeverycluster
6. While$(\mathbf{i<I_{max}})$
7. Forevery cluster$\mathbf{C_i}$
8. Executehuntingroutineforchaserfish using equation :

$$\boldsymbol{Chaser_{position}=Chaser_{position} + \alpha \oplus Levy(\beta)}$$

$$\boldsymbol{\beta = 0} < \beta \leq 2 \; and \; \alpha = 1$$

9. Executeblockingroutineforblockerfish

$$\boldsymbol{Blocker_{position}=D_g * e^{b\rho} * Cos2\pi\rho + Chaser_{position}}$$

$$where \; \rho \; is \; a \; random \; number \; among \; [a, 1] \; and \; b = 1$$

$$a \; is \; linearly \; decreased \; from \; -1 \; to \; -2 \; as \; iterations \; increase$$

$$\boldsymbol{D_g = r * Chaser_{position} - Blocker_{position}}$$

$$where\ r\ is\ a\ random\ number\ among\ [-1,1]$$

10. Calculatethefitnessvalueofeachgoatfish positions
11. If$\boldsymbol{Blocker_{fitness}}$hasbetterfitnessvaluethan$\boldsymbol{Chaser_{fitness}}$
12. exchangerolesbyupdating $\boldsymbol{Chaser_{fitness}}$
13. EndIf
14. If$\boldsymbol{Chaser_{fitness}}$hasbetterfitnessvaluethan$\boldsymbol{g_{fitness}}$
15. Update $\boldsymbol{g_{bfitness}}$
16. EndIf
17. If the fitnessvalue of$\boldsymbol{Chaser_{fitness}}$hasnotimproved
18. $q \leftarrow q+1$
19. EndIf
20. If $q>\lambda$
21. Executea routineforchangingzone using PSO Algorithm

$$\boldsymbol{V = w * V + c_1 * r_1(Chaser_{bposition} - Position_{ts}) + c_2 * r_2(g_{bposition} - Position_{ts})}$$

$$\boldsymbol{Position_{ts} = Position_{ts} + V}$$

$$where\ \boldsymbol{r_1, r_2}\ is\ a\ random\ number\ among\ [0,1]$$

$$where\ acceleration\ coefficients\ \boldsymbol{c_1, c_2}\ is\ equal\ 2\ and\ inertia\ weight\ \boldsymbol{w}\ is\ 0.4$$

22. $q \leftarrow 0$
23. EndIf
24. End For
25. i←i+ 1
26. End While
27. Output$\boldsymbol{g_{bposition}}$

After calculating the fitness value, the HPSO model selects the global best fitness value and population is separated into k clusters. After this, the position of chase and blocker fish for every cluster is evaluated. For hunting the chaser fish utilizes below given equation;

$\boldsymbol{Chaser_{position}=Chaser_{position} + \alpha \oplus Levy(\beta)}$ ----------**(3)**

$$\boldsymbol{\beta = 0 < \beta \leq 2}\ and\ \alpha = 1$$

On the other hand, the blocker fish utilizes the equation given below;

$\boldsymbol{Blocker_{position}=D_g * e^{b\rho} * Cos2\pi\rho + Chaser_{position}}$ ----------**(4)**

Wherein ρ represents the random number in between [a.1] and b=1. The value of $\boldsymbol{D_g}$can be calculated by using the below given equation.

$\boldsymbol{D_g = r * Chaser_{position} - Blocker_{position}}$----------**(5)**

Where, r represents the random number with range from [-1 1] respectively. Compute value for each goatfish position. The value of $\boldsymbol{Chaser_{fitness}}$is updated if$\boldsymbol{Blocker_{fitness}}$is greater than$\boldsymbol{Chaser_{fitness}}$. Otherwise, value of$\boldsymbol{g_{fitness}}$is updated when$\boldsymbol{Chaser_{fitness}}$has better fitness value. In addition to this, PSO algorithm has been used in the proposed model for changing the zones in case when the value of q

is greater than λ.The equation utilized for analyzing the value of v is given below;

$$V = w * V + c_1 * r_1(Chaser_{bposition} - Position_{ts}) + c_2 * r_2(g_{bposition} - Position_{ts})\text{----------(6)}$$

$$Position_{ts} = Position_{ts} + V\text{----------(7)}$$

Where r_1, r_2 represents a random number between 0 and 1 and c_1, c_2 represents the acceleration coefficients whose value is 2 and w represents the inertia weight whose value is 0.4 respectively. This process is repeated for fixed number of iterations and finally output is generated that determines how tasks need to be scheduled.

d) Performance evaluation

Once the fitness is calculated and tasks are scheduled in cloud computing, Now comes the ideal moment to evaluate how well the suggested Hybrid YPSO model performs for a given fitness values. By contrasting the suggested model's efficacy with that of a conventional YSGA algorithm in terms of fitness values for various scheduling tasks, the proposed model's effectiveness has been evaluated and proven. The detailed discussion of the results attained is given in the next phase of this manuscript.

4. Experimental Results and Discussion

By contrasting the suggested Hybrid YPSO approach with the conventional YSGA model in two scenarios involving different tasks and virtual machines, respectively, the ability of the proposed framework is examined and assessed. The simulation outcomes were obtained in terms of their fitness value for different tasks with 3 VMs and for varying VMs. In this section of paper, we are going to elaborate the results obtained for proposed Hybrid YPSO model and traditional YSGA models.

a) Performance Evaluation with varying tasks and 3 VMs

In the very beginning we have analyzed the performance of proposed model by comparing it with traditional YSGA model in terms of their fitness value for 10 tasks and 3 VMs. The graph obtained for the same is shown in figure 2. From the given graph, it is observed that in standard YSGA model the fitness value is initially 137 which then goes decreasing and reaches to a minimum value of 126. On the other hand, the fitness value is optimized in proposed hybrid YPSO model to 117. Moreover, initially, the fitness value in proposed hybrid YPSO model was only 125 which then go to decreasing with the increase in number of iterations.

Furthermore, proposed system has also analyzed and validated the performance of proposed hybrid YPSO model by contrasting it with standard YSGA model in terms of their fitness for 20 tasks with three VMs. The comparison graph obtained for the same is shown in figure 3. After analyzing the above given graph the fitness value attained in standard YSGA based task scheduling model was 222.95 whereas, it was 212.28 in proposed hybrid YPSO model.

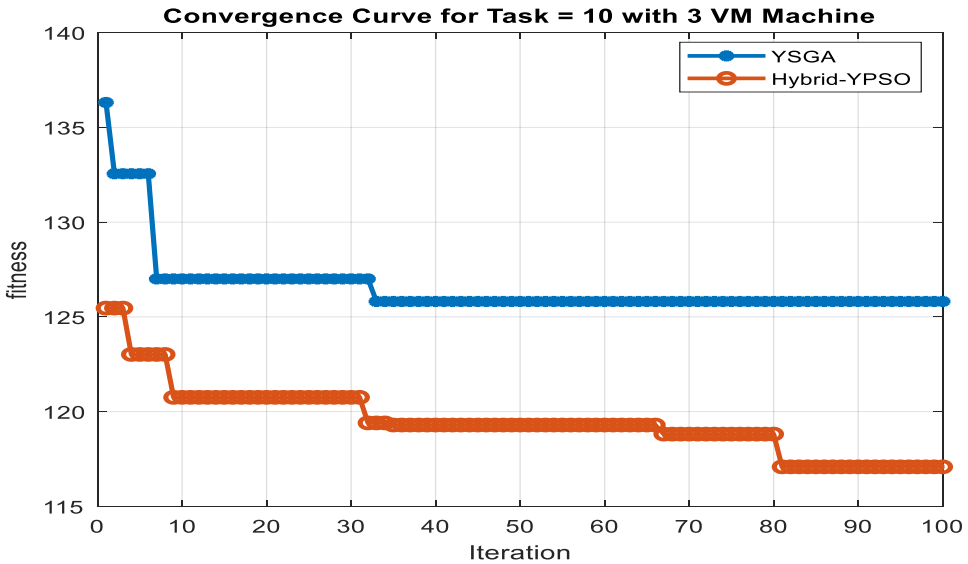

Fig.2. Comparative graph for fitness with 10 tasks

This decrease value of fitness in proposed hybrid YPSO model for scheduling 20 tasks simultaneously determines its effectiveness over standard YSGA approach.

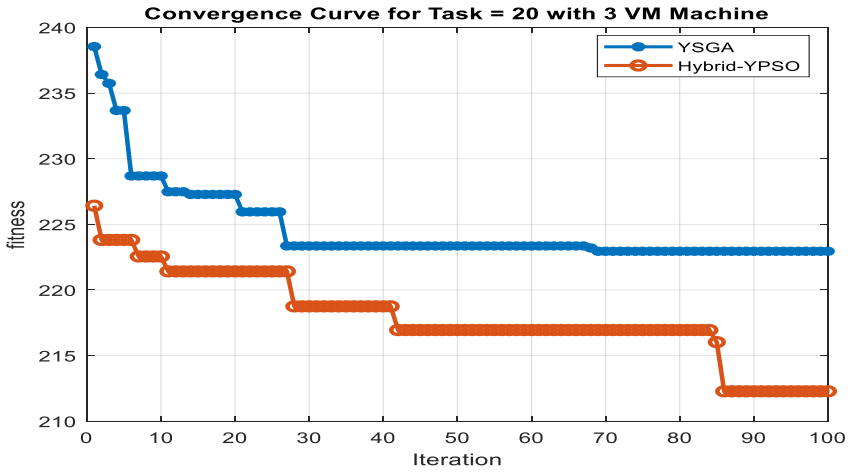

Fig. 3. Comparative graph for fitness with 20 tasks

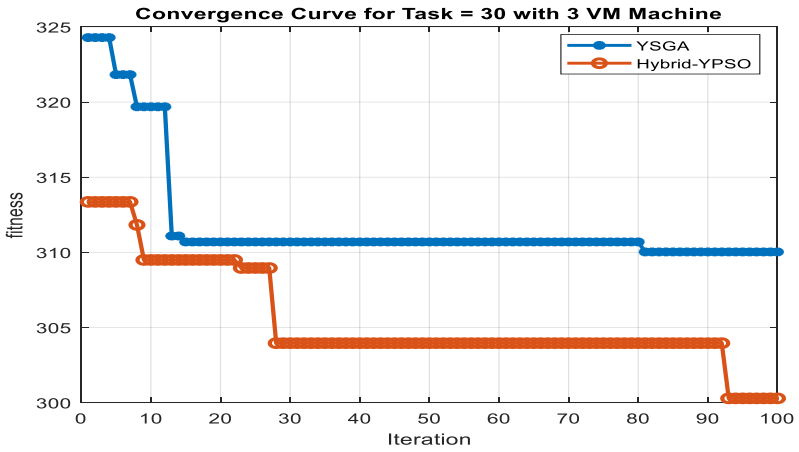

Fig.4. Comparative graph for fitness with 30 tasks

Figure 4 illustrates the comparative graph attained for fitness curve when total number of tasks are 30 and 3 VMs are used. Total iterations and the corresponding fitness values are shown by the x-axis and y-axis in the provided graph, respectively. The graph shown above reveals that value of fitness curve was reduced to minimum of 310 in standard YSGA model. However, the fitness curve was further reduced and optimized to a value of 300 in proposed hybrid YPSO model with respect to number of iterations. This shows that proposed hybrid YPSO model is scheduling tasks more effectively than traditional YSGA model.

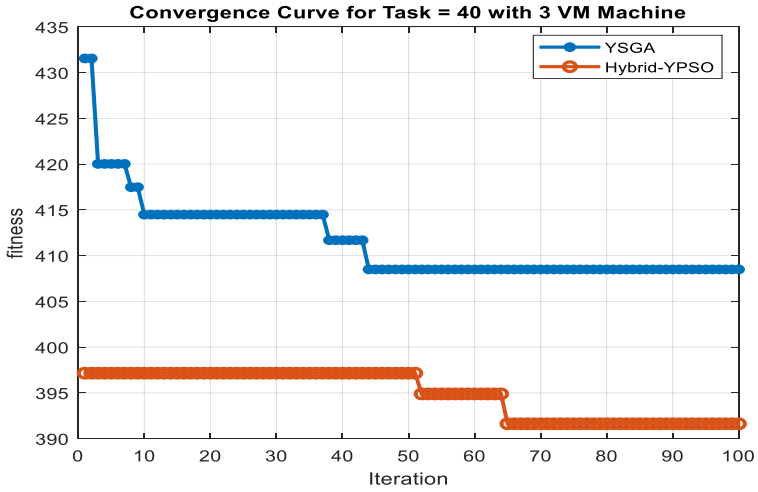

Fig.5. Comparative graph for fitness with 40 tasks

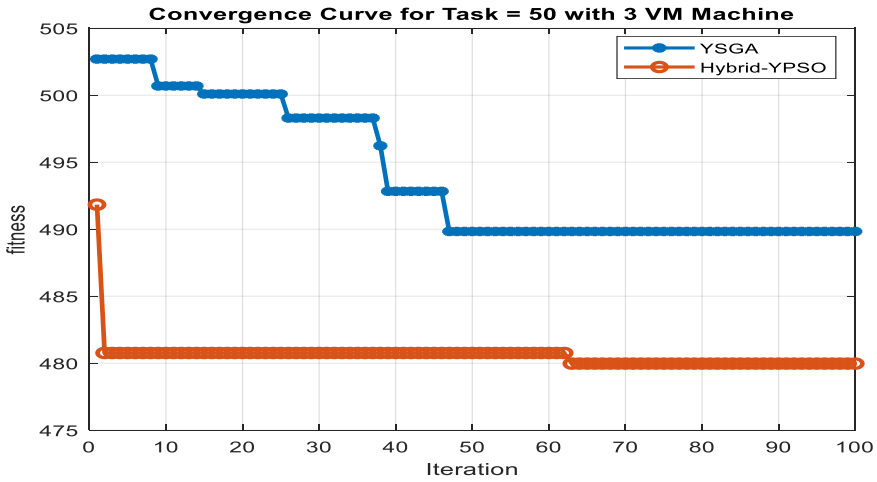

Fig. 6. Comparative graph for fitness with 50 tasks

Similarly, we have also analyzed the efficacy and efficiency of proposed hybrid YPSO model in terms of its fitness curve for 40 and 50 tasks respectively. Figures 5 and 6 illustrate the comparison graphs that are created for the same. After analyzing the above two graphs, it can be concluded that in both cases the proposed hybrid YSPSO model is showing least fitness curve then standard YSGA model. On the contrary side, the fitness curve was highest in standard YSGA model even though with increasing number of iterations the fitness curve is still high than proposed hybrid YPSO model.

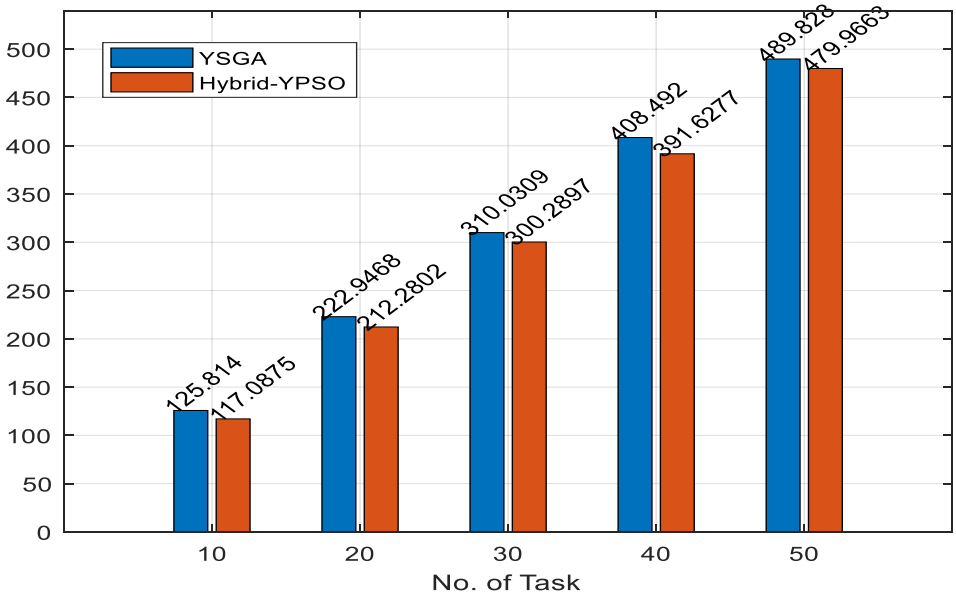

Fig.7. Comparison graph for fitness on different tasks

In order to know the exact values of fitness curves attained by proposed hybrid YPSO model and standard YSGA model, we have determined the fitness values for different tasks of 10, 20, 30, 40 and 50 with 3 VMs through bar graph that is shown in figure 7. The blue and orange bar depicts the performance of standard YSGA and hybrid YPSO model respectively. From the given graph, it is observed that for 10 tasks, the fitness curve was 125 in traditional YSGA model whereas; it was only 117 in proposed hybrid YPSO model. Likewise, when number of tasks were raised to 20, the fitness curve in standard YSGA was 222 while as, it was only 212 in proposed hybrid YPSO model. Also, when number of tasks were raised to 30, 40 and 50, the fitness curve in standard YSGA model was 310, 408 and 489 respectively, showing increased value with number of tasks increase. On the other end, the fitness value came out to be only 300, 391 and 479 in proposed hybrid YPSO model with 30, 40 and 50 number of tasks allocated. The specific value of fitness curves attained in both models is recorded in tabular form also and is shown in table 2.

Table 2: Specific value for fitness for different tasks

Task	YSGA	Hybrid YPSO
10	125.81	117.09
20	222.95	212.28
30	310.03	300.29
40	408.49	391.63
50	489.83	479.97

b) Analysis with varying VMs

In second phase of work, we have analyzed and examined the performance of standard YSGA and proposed Hybrid YPSO models in terms of fitness for varying VMs. The main reason for doing so is to ensure how well our model performs when VM count is increased in proposed model. However, before analyzing the performance of proposed hybrid YPSO with varying VMs, we have tried to analyze the fitness trend in standard YSGA model with varying VMs. The analytical graph obtained for standard YSGA with changing VMs is depicted in Fig 8. The x and y axis of the given graph depicts the varying VMs and their fitness value. After carefully reviewing the graph, we observed that with the increase in VMs count the fitness curve tends to decrease in standard YSGA model for different tasks. However, the fitness attained in this model is not efficient and hence we have tried to optimize it in our proposed hybrid YPSO approach.

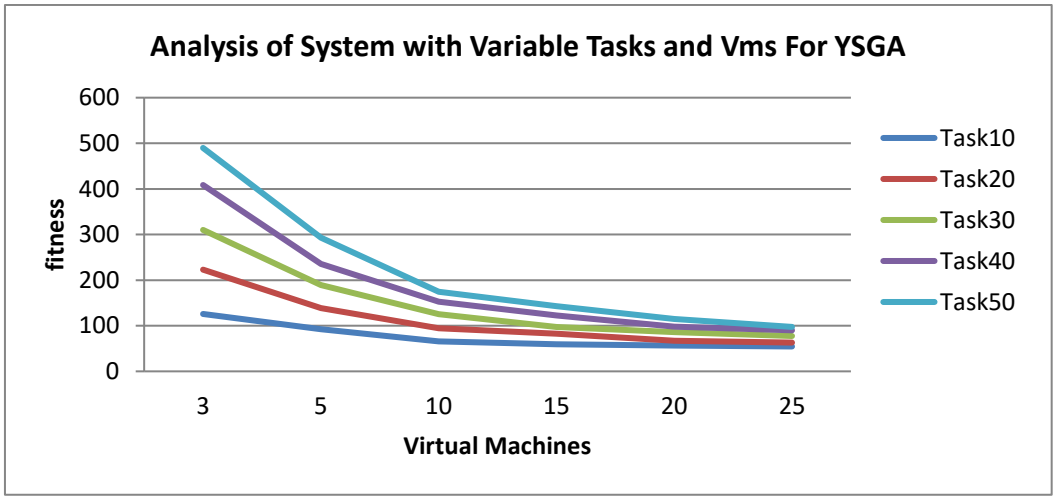

Fig 8. Fitness graph obtained for standard YPSO

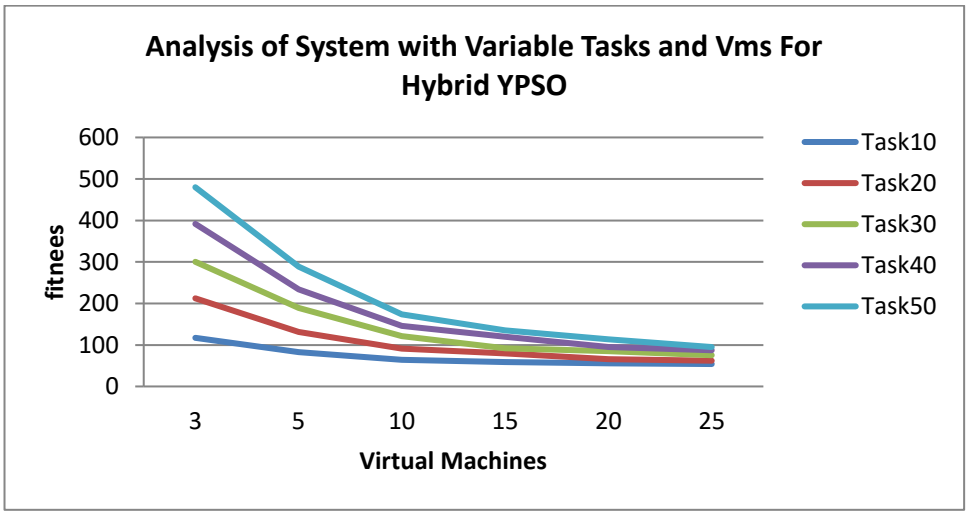

Fig 9. Fitness graph obtained for proposed YPSO

Next, in order to analyze the effectiveness of proposed hybrid YPSO model, we examined its fitness function under varying VM count. The analytical graph obtained for proposed hybrid YPSO for increasing VM count is

depicted in Fig 9. After carefully analyzing the graph, we observed that our proposed model follows the same trend of decreasing fitness value with increase in VM count, as it was followed by standard YSGA model. The only difference between the two models is that our hybrid YPSO model is able to reduce fitness value more effectively than standard YSGA model. This shows that by integrating YSGA and PSO algorithms, we tend to achieve lower fitness values which is the main aim of our work.

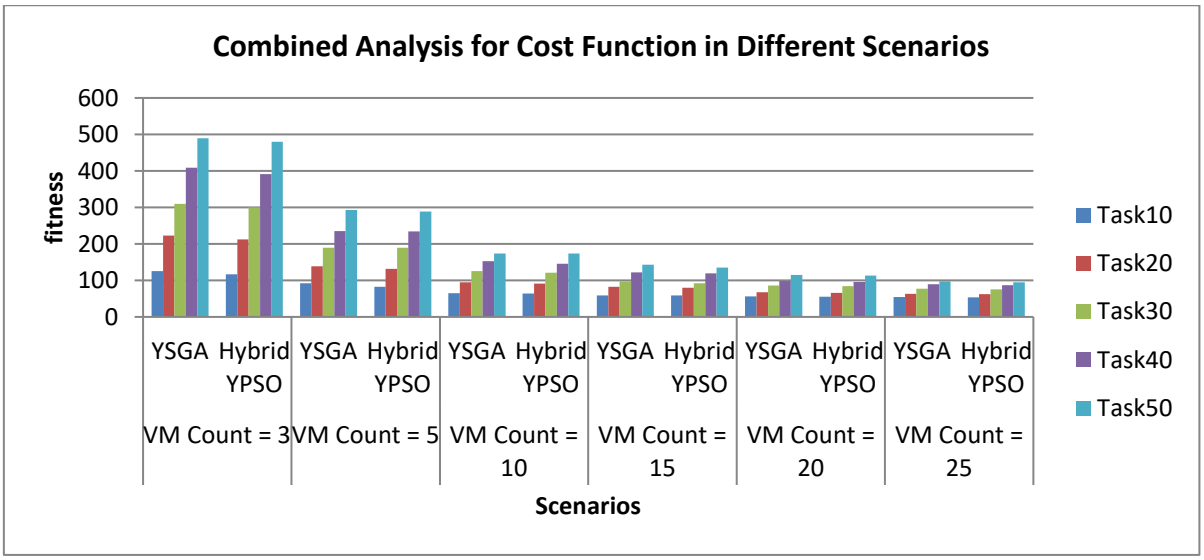

Fig 10. Comparative analysis for fitness with varying VMs

Moreover, to validate and prove the effectiveness of proposed hybrid YPSO approach in terms of its fitness value for varying VM count, we compared its fitness value with standard YSGA model. The comparative graph obtained for the same is shown in Fig 10. The horizontal axis determines different models and VM size whereas, vertical axis determines fitness values for different tasks respectively. Through carful analysis, we observed that when VM count was 3, the fitness value was 117.09 for 10 tasks and 479.9 for 50 tasks in proposed HPSO model whereas, it was 125.8 and 489.8 in standard YSGA model. Likewise, the fitness value came out to be 82.8 and 288 for 10 and 50 tasks with 5 VMs respectively. Also, the fitness value came out to be 135.2, 113 and 94 in proposed HPSO model with 15, 20 and 25 VMs and 50 tasks whereas, it was 142.7, 114.9 and 97.4 in standard YSGA model respectively. The specific value of fitness for different VMs and tasks are specified in Table 3.

Table 3: Comparative Analysis of Fitness with varying VMs

VM Count	Task	10	20	30	40	50
3	YSGA	125.81	222.95	310.03	408.49	489.83
	Hybrid YPSO	117.09	212.28	300.29	391.63	479.97
5	YSGA	92.55333333	138.56	189.244	235.3614	293.4114019
	Hybrid YPSO	82.838	131.6073	189.2470769	233.8931	288.5653211
10	YSGA	65.40615385	94.76769	125.5894737	152.4957	174.1878571
	Hybrid YPSO	64.46	91.17478	121.5525714	145.8333	173.6253571
15	YSGA	59.1569697	82.30116	97.46820513	122.4154	142.7734752
	Hybrid YPSO	58.45515152	79.494	92.06909091	119.2988	135.2550388
20	YSGA	56.65666667	67.23846	86.36291667	98.04	114.9702857
	Hybrid YPSO	55.62636364	65.91692	84.71863636	95.3052	113.3934286
25	YSGA	54.50363636	63.07086	77.51180952	89.97289	97.45555556
	Hybrid YPSO	53.95236364	62.03046	75.28977778	87.22667	94.9464

After carefully reviewing the results, we observed that proposed hybrid YPSO model is outperforming traditional YSGA model in both scenarios. The findings conclude that even through with the increase in number of tasks the fitness value also increases but still the proposed model is showing effective fitness results than standard YSGA model. Furthermore, in the second case, where we are varying the VMs the proposed model is again showing better results, as we observed that with the increase in VM count the fitness values goes on decreasing, indicating the efficacy of proposed model. The results clearly indicate that even though number of tasks are increasing along with VM count, the fitness only keeps getting better.

5. Conclusion

This paper presents an effective and optimized task scheduling model in which two optimization algorithms i.e. YSGA and PSO are hybridized. The efficacy and efficiency of proposed hybrid YPSO model is evaluated and analyzed in MATLAB software. The experimental outcomes were determined and compared with standard YSGA model in terms of their fitness values with respect to varying tasks. After analyzing the results, it is observed that proposed hybrid YPSO model has least fitness value for different tasks than standard YSGA model. The value of fitness was 125 in standard YSGA and 117 in proposed hybrid YPSO model for 10 tasks. However, with the increase in number of tasks to 20, 30 40 and 50, the fitness value also increases. The value of fitness was 222, 310, 408 and 489 in traditional YSGA model for 20, 30, 40 and 50 tasks. While as, in proposed hybrid YPSO model the fitness function was still least at 212, 300, 391 and 479 for 20, 30, 40 and 50 tasks. Moreover, when the performance of proposed model is analyzed for varying VMs, we observed that fitness value is getting reduced. Results showcased that proposed Hybrid YPSO model attains fitness of 117.09, 82.8, 64.4, 58.4, 55.6 and 53.9 for 3, 5, 10, 15, 20 and 25 VMs with 10 tasks respectively, however, it was 125.8, 92.5, 65.4, 59.1, 56.6 and 54.5 for given VMs respectively. Similarly, fitness value was reduced by around 10%, 5%, 1%, 7%, 1% and 3% in proposed hybrid YPSO model for 3, 5, 10, 15, 20 and 25 VMs with 50 tasks respectively. This shows that even through the tasks and VMs count is increasing, the fitness value is still low in proposed hybrid YPSO model than other similar approaches.

References

[1] Niranjan Mani Tripathi. (2015). Load balancing in cloud computing: review, taxonomy and future directions. Journal of Network and Computer Applications, 56, 11-23.

[2] Mani Tripathi, N. (2017). Load balancing in cloud computing: a review. Journal of Parallel and Distributed Computing, 97, 3-15.

[3] Niranjan Mani Tripathi. (2018). Task scheduling in cloud computing: a review. Journal of Cloud Computing: Advances, Systems and Applications, 7(1), 1-15.

[4] Saeed, F. et al. (2019). Load Balancing on Cloud Analyst Using First Come First Serve Scheduling Algorithm. In: Xhafa, F., Barolli, L., Greguš, M. (eds) Advances in Intelligent Networking and Collaborative Systems. INCoS 2018. Lecture Notes on Data Engineering and Communications Technologies, vol 23. Springer, Cham. https://doi.org/10.1007/978-3-319-98557-2_42

[5] Tarandeep, Bhushan, K. (2020). Load Balancing in Cloud Through Task Scheduling. In: Sharma, H., Pundir, A., Yadav, N., Sharma, A., Das, S. (eds) Recent Trends in Communication and Intelligent Systems. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-15-0426-6_21

[6] Pradhan, A., &Bisoy, S. K. (2022). A novel load balancing technique for cloud computing platform based on PSO. Journal of King Saud University - Computer and Information Sciences, 34(7), 3988-3995. https://doi.org/10.1016/j.jksuci.2020.10.016

[7] Priya, V., Sathiya Kumar, C., & Kannan, R. (2019). Resource scheduling algorithm with load balancing for cloud service provisioning. Applied Soft Computing, 76, 416-424. https://doi.org/10.1016/j.asoc.2018.12.021

[8] N. Manikandan, N. Gobalakrishnan, and K. Pradeep, “Bee optimization based random double adaptive whale optimization model for task scheduling in cloud computing environment,” Computer Communications, vol. 187, pp. 35–44,2022.

[9] Attiya, I., Abd Elaziz, M. and Xiong, S., 2020. Job scheduling in cloud computing using a modified harris hawks optimization and simulated annealing algorithm. Computational intelligence and neuroscience, 2020.

[10] Zivkovic, M., Bezdan, T., Strumberger, I., Bacanin, N., Venkatachalam, K. (2021). Improved Harris Hawks Optimization Algorithm for Workflow Scheduling Challenge in Cloud–Edge Environment. In: Pandian, A., Fernando, X., Islam, S.M.S. (eds) Computer Networks, Big Data and IoT. Lecture Notes on Data Engineering and Communications Technologies, vol 66. Springer, Singapore.