Workflow Scheduling in Cloud Computing Environment by Combining Particle Swarm Optimization and Grey Wolf Optimization

Divya Makhija, Chapram Sudhakar, Posham Bhargava Reddy, and Varsha Kumari

National Institute of Technology Warangal, Warangal,
Telangana-506004, India

Abstract. Scheduling workflows is a vital challenge in cloud computing due to its NP-complete nature and if an efficient workflow task scheduling algorithm is not used then it affects the system's overall performance. Therefore, there is a need for an efficient workflow task scheduling algorithm that can distribute dependent tasks to virtual machines efficiently. In this paper, a hybrid workflow task scheduling algorithm based on a combination of Particle Swarm Optimization and Grey Wolf Optimization (PSO_GWO) algorithms, is proposed. PSO_GWO overcomes the disadvantages of both PSO and GWO algorithms by improving the exploitation (local search) of PSO algorithm and exploration (global search) of GWO algorithm. This leads to better balance between exploration and exploitation, consequently it minimizes the makespan with 5.52% compared to GWO and 3.68% compared to PSO. The degree of imbalance reduced upto 33.22% compared to GWO and 17.61% compared to PSO, improves the convergence rate as well depending on number tasks and iterations. CloudSim tool is used to evaluate the proposed algorithm. The simulation results confirmed that the proposed method performs better than both of the standard PSO and GWO in terms of makespan, degree of imbalance and convergence rate.

Keywords: Task Scheduling, Makespan, Particle Swarm Optimization, Grey Wolf Optimization, Exploration, Exploitation

1 Introduction

The need of cloud computing has increased a lot after the emergence of big data and availability of on-demand services in cloud computing. It allows users to access server, web applications etc. through internet. One of the most important operation of cloud computing is workflow task scheduling as it can create major effect on overall performance of the system [1]. In workflows tasks are not independent. Directed Acyclic Graph (DAG) [2] is used to represent the workflows. The nodes in the DAG indicate tasks and the directed edges which joins the nodes indicate the dependency between the tasks. Example of a workflow is given in Figure 1. It contains nine tasks t_1, t_2, ..., t_9. Edges in the DAG indicates dependency between tasks [3]. For example task t_3 is executed after task t_1, because t_3 is dependent on t_1.

Workflow task scheduling algorithms for cloud can be categorized as: heuristic algorithms and meta-heuristic algorithms [4]. On account that workflow task scheduling in a cloud environment is a famous NP-complete problem [5], therefore meta-heuristic algorithms [6] are the most carried out algorithms for task scheduling to find optimal or near optimal solutions in affordable time. Moreover, they include random choices to find the solutions.

In recent years several nature inspired optimization techniques [7] have been advanced. They encompass Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Ant Colony optimization (ACO), Grey Wolf Optimization (GWO) etc. The purpose of those algorithms is to find the optimal solution and improve the makespan as well. To do

DOI:10.5121/cseij.2022.12601

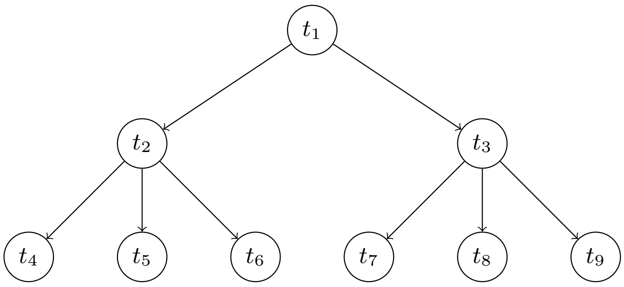

Fig. 1. Workflow schedule example

that researchers blended the nature-stimulated variant with exploration and exploitation to discover the optimal solution.

Exploration is the functionality of the version to search globally i.e., within the whole search space, whereas exploitation is the capability to search domestically i.e., close to the good result [8]. Subsequently, the intention of all nature stimulated variants is to make a balance between exploration and exploitation due to the fact that there is continually a tradeoff between these two. In the present work, a hybrid variation with the aid of combining PSO and GWO variations is presented and the performance of the proposed algorithm is compared with the existing PSO and GWO.

The rest of the paper is organized into the following sections. Section 2 discusses the related work with their pros and cons. The concepts of PSO, GWO, the proposed PSO_GWO algorithms and the formulation of task scheduling problem as an optimization problem are described in Section 3. Finally Section 4 and Section 5 presents experimental analysis and conclusion respectively.

2 Related Work

Workflow scheduling in cloud computing comes under NP-complete problem, so most of the researchers solved workflow task scheduling problem using heuristics and meta-heuristic techniques in the past. Choudhary et al. [9] proposed hybrid version of two algorithms - the first one is Gravitational Search Algorithm (GSA) and the second one is Heterogeneous Earliest Finish Time (HEFT) for bi-objective workflow scheduling. The merits of this algorithm are improved makespan and computational cost, and the demerits are for the tasks with high complexities, this algorithm may not give correct results. Tawfeek et al. [10] proposed task scheduling algorithm that is based on Ant Colony Optimization (ACO) to minimize the makespan but the algorithm is having slow convergence rate and it is taking more time to find feasible solutions.

Hamad and Omara [11] proposed task scheduling algorithm that is based on Genetic Algorithm (GA). This algorithm improves the makespan, cost and resource utilization but requires complex operators. A.Al-maamari, F. Omara [12] have designed a task scheduling algorithm that is a hybrid version of Cuckoo Search (CS) and PSO algorithm and named it as PSOCS, the target of this algorithm is to minimize makespan and utilize resources efficiently. Xu et al. [13] proposed task scheduling algorithm that is based on PSO. It includes less hyper parameters, reduces makespan and it is simple to implement but it has low convergence rate and exploitation is weak. Salman et al. [14] proved that PSO algorithm outperforms GA in terms of time and the quality of the solution as well.

A similar workflow scheduling problem has been addressed by Neeraj Arora, Rohitash K. Banyal [19] that is based on the standard hybridization model of PSO and GWO. It focuses on the analysis of total execution time and total execution cost. However, existing research focuses on the combination on two or more algorithms for better performance. In that hybrid approach of PSO and GWO algorithm, degree of imbalance and convergence rate are not considered. We formulate the hybridization of PSO and GWO that is based on a different approach in calculation of inertia weight (w) of PSO and controlling parameter (a) of GWO. It focuses on the analysis of average makespan time, degree of imbalance and convergence rate using different data set.

3 Proposed Algorithm

In this section brief introduction of PSO and GWO is given and the proposed PSO_GWO algorithm is described in detail.

3.1 Particle Swarm Optimization Algorithm

PSO algorithm is one of the meta-heuristic algorithms and is likewise called as population based stochastic algorithm founded by Dr. Eberhart and Dr. Kennedy in 1995. This algorithm mimics the social behavior of birds or school of fish [15]. In PSO a swarm of particles (potential solutions) looks for the global minimum value in the search space. Particles don't know about the position of global minimum, but they have fitness value and they try to optimize this fitness value to locate the global minimum solution.

Every particle in PSO is characterized by position and velocity vector and they update their position using velocity over the iterations to locate global minimum, and they maintain their personal best position and they also keep track of the global best position. So each particle knows its personal best position and also global best position.

Each particle updates their velocity and position using the following equations.

$$v_j^{t+1} = \omega * \left(v_j^t + c_1 r_1 \left(p_b^t - x_j^t\right) + c_2 r_2 \left(g_b - x_j^t\right)\right) \tag{1}$$

$$x_j^{t+1} = x_j^t + v_j^{t+1} \tag{2}$$

Here t is the current iteration, j is the current particle, v_j^t is the velocity of particle in the current iteration, p_b is the personal best position of the particle, g_b is the global best position of the particle, x_j^t is the position of the particle at time t and c_1, c_2 are acceleration coefficients. If c_1 is high, then particles focus only on their personal best solutions and if c_2 is high, then particles are more influenced by group rather than personal. r_1, r_2 are random numbers. ω is known as inertia weight. High value of ω results in exploration and low value of inertia weight results in exploitation, and ω is given as:

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) * (maxIter - t)/maxIter \tag{3}$$

Here ω_{min} is the minimum inertia weight, ω_{max} is the maximum inertia weight and $maxIter$ is the total number of iterations.

3.2 Grey Wolf Optimization Algorithm

Grey wolf optimizer is a population-based meta-heuristic algorithm that simulate the leadership ranking and hunting mechanism of gray wolves in nature. It is introduced by Seyedali Mirjalili et al. in 2014 [16].

The dominance of wolves is categorized into four types i.e., α, β, δ, and ω. α is taken into consideration as a pacesetter and the dominance decreases from α to ω. The lower degree wolves notify top level wolves. When this behavior is formulated as an algorithm then, α is considered as the remarkable answer, β is considered as the second-fine answer, δ is considered as the third-fine answer and the rest of the answers are considered as ω.

The GWO follows three steps to find the most suitable solution i.e., trying to find the prey, encircling the prey and attacking the prey. Mathematically the GWO algorithm can be modeled as described below. After searching the prey there is a phase called encircling the prey. The optimal equations for encircling the prey are given as:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X_p}(t) - \vec{X}(t)\right| \tag{4}$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A} \cdot \vec{D} \tag{5}$$

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a} \tag{6}$$

$$\vec{a} = 2 * (1 - t * t)/maxIter * maxIter \tag{7}$$

$$\vec{C} = 2 \cdot \vec{r}_2 \tag{8}$$

Where t is current iteration, $\vec{X}_p$ is the location of the prey, $\vec{X}$ is the location of solution of grey wolf. $\vec{A}$ and $\vec{C}$ are coefficient vectors and values of $\vec{A}$ range in between $[-2a, 2a]$, and the values of $\vec{C}$ range in between [0, 2], elements of $\vec{a}$ are reduced from 2 to 0 as the iteration number increases and $\vec{r_1}$, $\vec{r_2}$ are random vectors ranging between [0, 1].

After encircling, now is the turn for hunting. In GWO algorithm there are three best solutions i.e., α, β, δ that are used to determine the positions of the rest of the solutions. The equations for hunting phase are given as:

$$\overrightarrow{D_\alpha} = |\overrightarrow{C_1} \times \overrightarrow{X_\alpha} - \overrightarrow{X}| \tag{9}$$

$$\overrightarrow{D_\beta} = |\overrightarrow{C_2} \times \overrightarrow{X_\beta} - \overrightarrow{X}| \tag{10}$$

$$\overrightarrow{D_\delta} = |\overrightarrow{C_3} \times \overrightarrow{X_\delta} - \overrightarrow{X}| \tag{11}$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1} \times \overrightarrow{D_\alpha} \tag{12}$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2} \times \overrightarrow{D_\beta} \tag{13}$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3} \times \overrightarrow{D_\delta} \tag{14}$$

$$\overrightarrow{X}(t+1) = (\overrightarrow{X_1} + \overrightarrow{X_2} + \overrightarrow{X_3})/3 \tag{15}$$

Parameters $\vec{A}$ and $\vec{C}$ are used to balance between exploration and exploitation. When $|A| > 1$ and $C < 1$ exploration takes place and when $|A| < 1$ and $C < 1$ exploitation takes place.

3.3 Combined form of PSO and GWO

We have combined particle swarm optimizer and grey wolf optimizer i.e., the best features of both the variants are blended and in this way both the variants are running in parallel not one after another. This algorithm performs better than PSO and GWO because by combining these two the balance between exploitation and exploration is improved, i.e., it is improving the exploitation of PSO algorithm and the exploration of GWO algorithm. Consecutively it improves the makespan, convergence rate and degree of imbalance at the same time [17].

Mathematically combined approach can be modelled as described below. Position and velocity of particles are updated using below equations:

$$\mathrm{v}_j^{t+1} = \omega * \left[v_j^t + c_1 r_1 \left(X_1 - x_j^t\right) + c_2 r_2 \left(X_2 - x_j^t\right) + c_3 r_3 \left(X_3 - x_j^t\right)\right]$$

$$x_j^{t+1} = x_j^t + v_j^{t+1} \tag{16}$$

Where X_1, X_2, X_3 are the three best solutions which are calculated using Eq.(5).

Algorithm 1 PSO_GWO pseudocode

Randomly generate the initial population of N particles i.e., position and velocity of each particle X_j where $j = 1, 2, 3, 4, ..., N$
Initialize c_1, c_2, c_3, a, ω_{max}, ω_{min}
for $t \leftarrow 1$ to $maxIter$ **do** ▷ $maxIter$ = Total number of iterations
    update ω using Eq.(3)
    update a using Eq.(7)
    calculate fitness of each particle
    calculate three best solutions α, β and δ
    for $j \leftarrow 0$ to $N - 1$ **do**
        calculate A_1, A_2, A_3 using Eq.(6)
        calculate C_1, C_2, C_3 using Eq.(8)
        calculate $D_\alpha, D_\beta, D_\delta$ using Eq.(9), Eq.(10) and Eq.(11)
        calculate $X_\alpha, X_\beta, X_\delta$ using Eq.(12), Eq.(13), and Eq.(14)
        update velocity and position of current particle using Eq.(16) and Eq.(17)
        check for limits of position and velocity of current particle
    end for
    if $\alpha < globalBestFitness$ **then**
        $globalBestFitness \leftarrow \alpha$
    end if
end for

In the proposed algorithm in the first step the velocity and position are initialized randomly, which are of dimensions equal to the number of tasks for N particles or in other words the population consisting of N particles is initialized randomly. The next step is to initialize the parameters, after that the loop is executed for a specified number of iterations ($maxIter$). Inside this loop the steps as described in the Algorithm 1 are performed and after the loop terminates the calculated best solution (α) is compared with the current best solution ($globalBestFitness$). If α is better, then current best solution is updated.

3.4 Mapping of Task Scheduling Problem as an Optimization Problem

Every optimization algorithm consists of set of inputs, output, set of constraints and fitness function. Set of inputs are also known as variables and for given input values there is an output value. In every optimization problem there may be one or more sets of values for the inputs that gives the minimum value for the output. Constraints defines the limits that are applied on inputs. Fitness function takes input values and will give the estimate how close the given input values are to the optimal input values.

In the proposed algorithm main inputs are particles and every particle has velocity and position vector. The dimension of these vectors is equal to the number of tasks. The position vector denotes which task is assigned to which virtual machine. The output is minimum makespan and degree of imbalance of all the tasks. The constraints are maximum and minimum velocities that a particle can have and maximum and minimum positions. The fitness function consists of makespan and total execution time that includes the waiting time of the tasks.

4 Experiments and Performance Evaluation

4.1 Simulation Settings

For conducting the experiments CloudSim tool is used to simulate the proposed algorithm. The simulator is executed on a personal computer with Intel Core i5 10^th Generation, 8GB RAM, running Windows 11 operating system. The results of proposed algorithm i.e., combined version of PSO and GWO are compared against existing PSO algorithm and GWO algorithm. The parameters used in the simulation are given in Table 1 and the parameter values of PSO_GWO algorithm are given in Table 2.

Table 1. Parameter values of simulation

Parameter name	Value
Count of host	2
Count of VM	5
Count of iterations	1000
Population size	150
Count of tasks	30,50,70,90
MIPS of VMs	10-100

Table 2. Parameter values of PSO_GWO algorithm

Parameter name	Value
Inertia weight (ω)	0.9-0.4
c_1	2.0
c_2	2.0
c_3	2.0
a	2.0

4.2 Simulation Results

Each algorithm is executed 10 times and numbers of VMs kept as constant at five. The averages of the results are computed and these averages are taken into consideration for comparison. The graphical representation of comparison of average makespan of the proposed algorithm is shown in Figure 2. It can be observed from Figure 2 that the average makespan of proposed algorithm is reduced upto 5.52% compared to existing GWO algorithm and 3.68% less than that of the PSO algorithm.

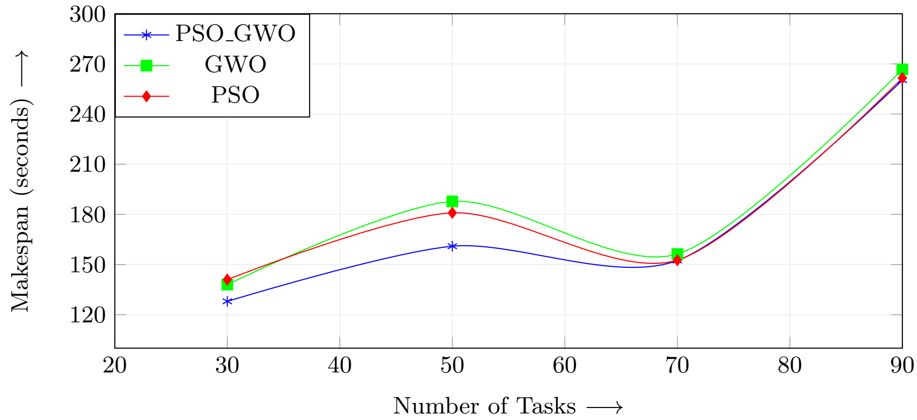

Fig. 2. Makespan for set of tasks

The load divided among the total number of VMs according to their computational capability, may lead to load imbalance which can be measured by the Degree of Imbalance (DI) [18]. DI considers three time factors and they are maximum execution time, minimum execution time, and average execution time. DI is calculated using the Eq. 18.

$$DI = \frac{t_{max} - t_{min}}{t_{avg}} \tag{17}$$

where t_{max}, t_{min} and t_{avg} are maximum execution time, minimum execution time and average execution time among all VMs respectively. The graphical representation of the comparison with PSO and GWO is given in Figure 3. From Figure 3 it can be observed that the proposed algorithm has minimum degree of imbalance among all i.e., 33.22% lower than GWO and 17.61% lower than PSO and less degree of imbalance means better balance in scheduling.

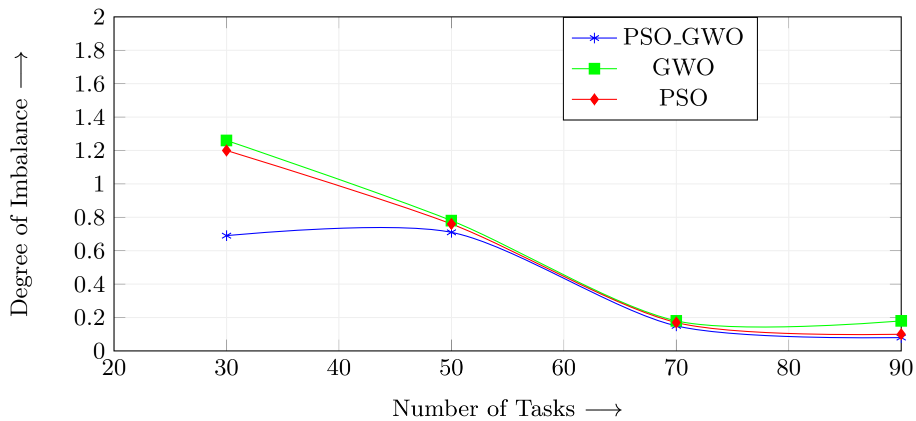

Fig. 3. Degree of imbalance for set of tasks

The result of convergence rate of PSO_GWO algorithm is compared against PSO and GWO algorithm. The comparison graphs of convergence rate of 30 tasks and 70 tasks are shown in Figure 4 and Figure 5 respectively. It can be observed from Figure 4 and Figure 5 that the convergence rate of PSO_GWO is better than PSO and GWO.

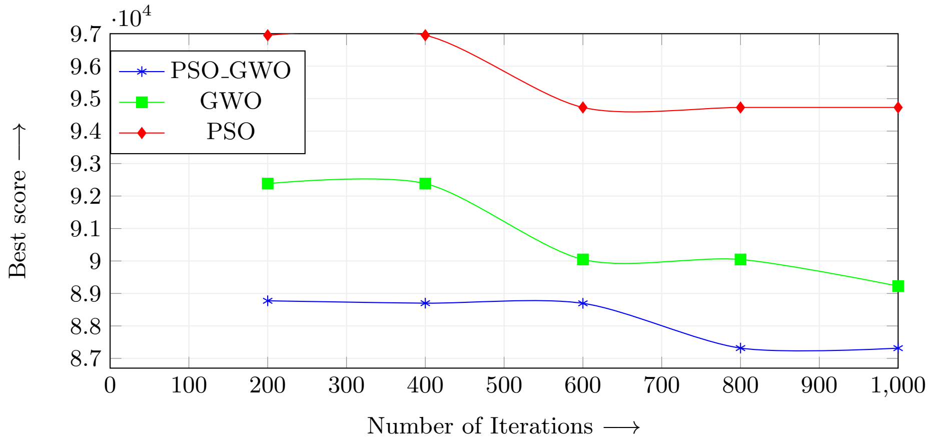

Fig. 4. Convergence rate for 30 tasks

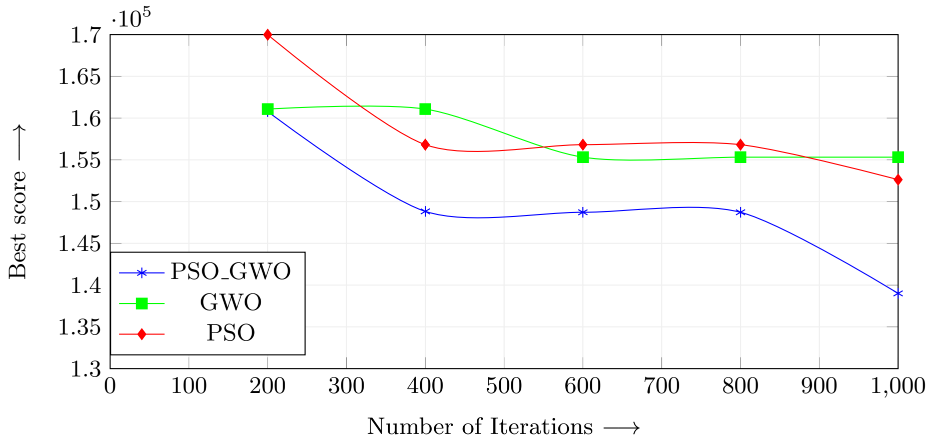

Fig. 5. Convergence rate for 70 tasks

5 Conclusion

In this paper, PSO_GWO task scheduling algorithm i.e., the hybrid version of PSO and GWO is presented. The simulation results shows that the hybrid task scheduling algorithm performs better than existing PSO and GWO in terms of makespan, degree of imbalance, and convergence rate. PSO_GWO performs better than PSO and GWO as it is making a better balance between exploitation and exploration than PSO and GWO. In future this work can be extended by considering the issue of deadlines and priorities of the tasks.

References

1. Peter Mell, Tim Grance, (2011), "The NIST definition of cloud computing", NIST, (2011).
2. Rafael Ferreira da Silva, Henri Casanova, Anne-Cécile Orgerie, Ryan Tanaka, Ewa Deelman, Frédéric Suter, (2020), "Characterizing, Modeling, and Accurately Simulating Power and Energy Consumption of I/O-intensive Scientific Workflows", Journal of Computational Science, Volume 44, 101157, ISSN 1877-7503, https://doi.org/10.1016/j.jocs.2020.101157.
3. Neeraj Arora and Rohitash Kumar Banyal, (2020), "HPSOGWO: A Hybrid Algorithm for Scientific Workflow Scheduling in Cloud Computing", International Journal of Advanced Computer Science and Applications(IJACSA), 11(10), http://dx.doi.org/10.14569/IJACSA.2020.0111078
4. V Vinothina, (2017), "Scheduling scientific workflow tasks in cloud using swarm intelligence", In 2017 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), IEEE, pp. 1–5.

5. Jeffrey D Ullman, (1975), "NP-complete scheduling problems", Journal of Computer and System sciences 10, 3 (1975), pp. 384–393.
6. Shantanu A. Lohi, Nandita Tiwari, Varsha Namdeo, and Snehal A. Lohi, (2020), "Analysis and review of effectiveness of metaheuristics in task scheduling process with delineating machine learning as suitable alternative", In 2020 International Conference on Innovative Trends in Information Technology (ICITIIT), pp. 1–6. https://doi.org/10.1109/ICITIIT49094.2020.9071559
7. S. H. Adil, K. Raza, U. Ahmed, S. S. A. Ali and M. Hashmani, (2015), "Cloud task scheduling using nature inspired meta-heuristic algorithm", 2015 International Conference on Open Source Systems Technologies (ICOSST), pp. 158-164, doi: 10.1109/ICOSST.2015.7396420.
8. E Emary and Hossam M Zawbaa, (2016), "Impact of chaos functions on modern swarm optimizers", PloS one 11, 7, e0158738.
9. Anubhav Choudhary, Indrajeet Gupta, Vishakha Singh, Prasanta K. Jana, (2018), "A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing", Future Generation Computer Systems, Volume 83, pp. 14-26, ISSN 0167-739X, https://doi.org/10.1016/j.future.2018.01.005.
10. Medhat A. Tawfeek, Ashraf El-Sisi, Arabi E. Keshk, and Fawzy A. Torkey, (2013), "Cloud task scheduling based on ant colony optimization", In 2013 8th International Conference on Computer Engineering Systems (ICCES), pp. 64–69. https://doi.org/10.1109/ICCES.2013.6707172
11. Safwat Hamad and Fatma Omara, (2016), "Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment", International Journal of Advanced Computer Science and Applications, 7, (01, 2016), pp. 550–556.
12. Al-maamari, Ali Omara, Fatma., (2015), "Task Scheduling Using PSO Algorithm in Cloud Computing Environments" International Journal of Grid and Distributed Computing, 8, pp. 245-256, doi: 10.14257/ijgdc.2015.8.5.24.
13. Lili Xu, Kun Wang, Zhiyou Ouyang, and Xin Qi, (2014), "An improved binary PSO-based task scheduling algorithm in green cloud computing", In 9th International Conference on Communications and Networking in China, pp. 126–131. https://doi.org/10.1109/CHINACOM.2014.7054272.
14. Ayed Salman, Imtiaz Ahmad, and Sabah Al-Madani, (2002), "Particle swarm optimization for task assignment problem", Microprocessors and Microsystems, 26, 8 (2002), pp. 363–371.
15. Daqin Wu, (2018), "Cloud computing task scheduling policy based on improved particle swarm optimization", In 2018 International Conference on Virtual Reality and Intelligent Systems (ICVRIS), IEEE, pp. 99–101.
16. A. Alzaqebah, R. Al-Sayyed and R. Masadeh, (2019), "Task Scheduling based on Modified Grey Wolf Optimizer in Cloud Computing Environment", 2nd International Conference on new Trends in Computing Sciences (ICTCS), pp. 1-6, doi: 10.1109/ICTCS.2019.8923071.
17. Narinder Singh and SB Singh, (2017), "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance", Journal of Applied Mathematics, pp. 1-15.
18. Dhinesh Babu LD and P Venkata Krishna, (2013), "Honey bee behavior inspired load balancing of tasks in cloud computing environments", Applied soft computing, 13, 5 (2013), pp. 2292–2303.
19. Neeraj Arora, Rohitash K. Banyal, "Workflow scheduling using particle swarm optimization and gray wolf optimization algorithm in cloud computing", In Concurrency and Computation: Practice and Experience 33.16, Wiley(2021).https://doi:10.1002/cpe.6281.

Authors

Divya Makhija, She received M.Tech from National Institute of Technology Warangal. She did Bachelor degree in computer science from Sagar institute of research and technology. Currently, she is working in ServiceNow. Her research interests include Soft Computing and Cloud Computing.

Chapram Sudhakar, He received the Ph.D. degree from National Institute of Technology Warangal, India, in 2011. He is currently Professor in the Department of Computer Science and Engineering, National Institute of Technology, Warangal, India. His research interests

include Unix Internals, cloud computing, distributed systems and high-performance computing.

Posham Bhargava Reddy, He received M.Tech from JNTUA College of Engineering Ananthapuramu. He did Bachelor degree in Information Technology. Currently, he is pursuing his PhD in Computer Science and Engineering department from National Institute of Technology Warangal. His research interests include Cloud computing, Fog Computing Task Scheduling, Resource allocation, an Load balancing.

Varsha Kumari, She received M.Tech from Aligarh Muslim University. She did Bachelor degree in Information Technology. Currently, she is pursuing her PhD in Computer Science and Engineering department from National Institute of Technology Warangal. Her research interests include Cloud computing, Fog Computing.