Volume: 16 Issue: 05 Pages: 6544-6550 (2025) ISSN: 0975-0290

Performance Analysis and Design f Task Scheduling Algorithms With Cloud Computing Technology

Aslan Samet BİLGİÇ
Bilgisayar Mühendisliği, Tokat Gaziosmanpaşa Üniversitesi
E-posta: aslansamet.bilgic6822@gop.edu.tr
Kenan ZENGİN
Bilgisayar Mühendisliği, Tokat Gaziosmanpaşa Üniversitesi
E-posta: kenan.zengin@gop.edu.tr

ABSTRACT

Cloud computing has become a current and popular technology in recent years and has come across us in every field. In fact, the fact that it comes across us in every field shows why this technology is popular. Today, many devices do not have sufficient resources despite having an internet connection. What we mean by resources here is that the processing ability, storage space and energy source are not sufficient. This is where cloud computing comes into play to solve these problems. Devices with low resources can also access large data and the high complexity calculations it requires. We can define cloud computing as an internet-based computing system that provides adaptive computing resources, storage areas, different applications and servers without the need for interaction with the service provider and with minimum management cost. Systems and commercial service services that provide services related to Cloud Computing and open source systems such as OpenStack have been researched and CloudSim has been used for research studies. CloudSim is a simulator that includes the infrastructure and services of open source cloud computing. It was developed in Java by CLOUDS Lab. Java is an object-oriented language, which provides researchers with ease of use in this sense. We use task scheduling algorithms in CloudSim with 30 inputs and their process performance was examined.

Keywords - Cloudsim, Cloud Computing, Grey Wolf Optimization(gwo), Particle Swarm Optimization (pso) and Task Scheduling

Date of Submission: December 25, 2024 Date of Acceptance: January 21, 2025

I. INTRODUCTION

Cloud computing has become a current and popular technology in recent years and has come across us in every field. In fact, the fact that it comes across us in every field shows why this technology is popular. As it is, many devices today do not have sufficient resources despite having an internet connection. What we mean by resources here is that the processing ability, storage space and energy source are not sufficient. This is where cloud computing comes into play to solve these problems. Large data and the high complexity calculations it requires can be accessed on devices with low resources. If we define Cloud Computing, it is an internet-based computing system that provides adaptive computing resources, storage areas, servers, different applications and services without the need for interaction with the service provider and with minimum management cost. [1] Therefore, Cloud Computing is a promising technology. In our research, the performances of Cloud Computing in task scheduling algorithms will be examined. In the study, the task scheduling optimization of Cloud Computing was examined using heuristic algorithms. Cloud computing-related services and commercial service services and open source systems such as OpenStack have been investigated. CloudSim was used for research studies. CloudSim is a simulator that includes the infrastructure and services of open source cloud computing. It was developed in Java by CLOUDS Lab. Being developed in Java is also an advantage for developers who develop software with Java. Since Java is an object-oriented language, it provides ease of use to researchers in this sense. When examining the performance of Cloud Computing in task scheduling algorithms, attention should be paid to three factors in the resource usage of computers. Processor-intensive, Storage-intensive and Network traffic-intensive usage should be organized well. Namely, when a processor-intensive usage occurs, there will be competition in processor sockets and other resources (memory, disk, etc.) will be wasted without being used. As a result, the service quality will not be high and bottlenecks will occur. Therefore, since the usage of other resources will be low, energy will be lost due to the wasting of some resources. Task scheduling is a technique used to match the tasks of clients with the available and suitable virtualized resources using an efficient algorithm [2]. In heterogeneous computing such as cloud computing, the task scheduling problem becomes more difficult as it is a distributed and scalable environment. Therefore, an effective task scheduling algorithm is needed, which is considered as the key to the performance of the system [2-3]. In a cloud system, there are three common categories of task scheduling algorithms [4]. These are;

• Traditional algorithms such as First Come First Serve (FCFS), Shortest Job First (SJF), Longest Job First (LJF) and Round Robin(RR) [5]

• Heuristic algorithms such as Min-Max and Max-Min algorithms [6]

• Metaheuristic algorithms such as Ant Colony Optimization Algorithm (ACO) [7] and Particle Swarm Optimization (PSO) [8].

Generally, a common 300W server consumes about 2628KWH of energy per year, while an additional 748KWH of electricity is needed for cooling [9-10]. One of the factors that make cloud computing economical is that it saves energy costs by reducing energy consumption, and reducing costs without compromising the basic criteria of performance and security is important for the sector. A suitable algorithm is needed to use our resources efficiently by reducing energy costs without creating any security or performance vulnerabilities. Scheduling refers to the way processes are assigned to run on the existing Central Processing Unit (CPU) [11]. Certain CPU scheduling techniques or algorithms have been developed. The general purpose of these algorithms is to use the CPU at the highest level of efficiency, to minimize average waiting time (AWT) and average turnaround time (ATAT), and to increase efficiency. The algorithms that will be used in our study are as follows:

- First Come First Serve (FCFS)
- Shortest Job First (SJF)
- Round Robin (RR)
- Particle Swarm Optimization (PSO)
- Grey Wolf Optimization (GWO)

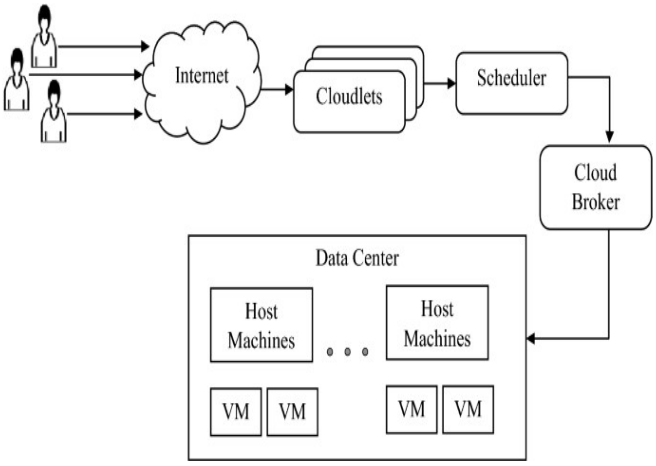

Fig. 1.1[25]

II. Method and Related Works

CloudSim is a simulator that includes the infrastructure and services of open source cloud computing. It was developed by CLOUDS Lab in Java. It is a good alternative for conducting research and academic studies on cloud systems. Although there are many cloud computing environments, open source systems such as OpenStack, which are free and paid, are complex and difficult to start with, while Cloudsim is developed in Java and Java is an object-oriented and widespread language, which provides ease of use to researchers in this sense. CloudSim has a two-layer structure.

1. CloudSim layer; It is the layer where the creation and management of basic assets such as Virtual Machines, Cloudlets, Hosts, and resources, as well as network-related execution are provided.

2. User Code layer; It is the layer where hardware features and requirements are created according to the scenario controlled by the user.

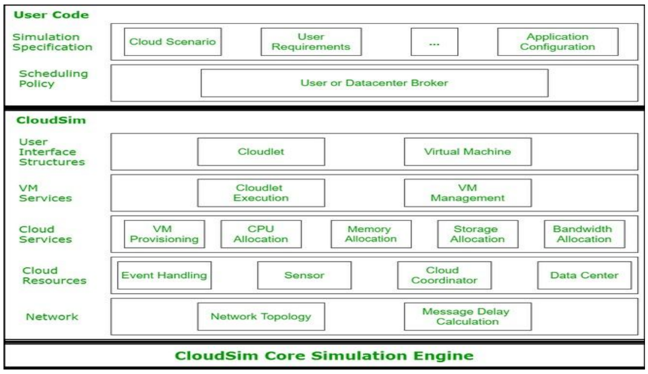

fig 2.1 [26]

Some of the most common classes used during simulation are:

- Datacenter: It is used to model the basic hardware equipment of any cloud environment, namely the Data Center. This class provides methods to determine the functional requirements of the Data Center as well as methods to determine the allocation of Virtual Machines.
- Host: This class performs the actions related to the management of virtual machines. It also performs the definitions for allocating CPU cores to virtual machines as well as providing memory and bandwidth to virtual machines.
- VM: This class represents a virtual machine by providing data that defines the bandwidth, RAM, and MIPS size of a Virtual machine, while also providing setter and getter methods for these parameters.
- Cloudlet: The Cloudlet class represents any task that is run on a virtual machine, such as a rendering task, memory access task, or a file update task. It provides methods similar to the Virtual Machines class, while also providing methods to define the execution time, status, cost, and history of a task. The data of this section is also used in the tests
- DatacenterBroker: Responsible for the operation of Virtual Machines, including creating, managing, destroying Virtual Machines and presenting cloudlets to the virtual machine.
- CloudSim: Responsible for starting the simulation environment after all required cloud assets are defined and then stopping it after all assets are destroyed.

2.1. First Come First Serve (FCFS)

The First Come First Serve algorithm is similar to the First In First Out structure in the queue data structure in terms of its working logic. It is an algorithm used in many areas, especially in operating systems in computer architecture. This algorithmic approach can be exemplified as follows. If you are at a hairdresser, the first person to arrive will be processed, and the others will wait for both this process and their turn to finish. Among those waiting, it does not matter whether the person has urgent work, short work or an important person, they have to wait their turn. It is an uninterrupted and primitive algorithm. Therefore, it causes

Volume: 16 Issue: 05 Pages: 6544-6550 (2025) ISSN: 0975-0290

situations such as CPU starvation, resulting in performance loss and inefficiency.

2.2-Shortest Job First (SJF)

It is a scheduling algorithm in which the shortest job is completed first. The jobs currently available are ranked according to the time required for completion. For this ordering, instead of a specific sorting algorithm, it is generally sorted primitively as a brute force, and the shortest-lasting job is ranked first, and the jobs are taken in order from smallest to largest. In this approach, which is an uninterrupted algorithm in terms of working logic, there is no chance for another job to intervene after a job starts. Performance is tried to increase by doing the shortest job at hand.

2.3-Round Robin Algorithm

Round Robin is an algorithm designed to be used in time-sharing systems. According to the algorithm, even if a process does not finish within the specified time, it is put on hold and a new process is started. Thus, a long process does not prevent other processes from being performed. The problem of not being able to access the resource needed by the process, which is called CPU starvation, is eliminated.

Data transfer is done in a circular order. For example, if the completion time of the 1st process is assumed to be 8 ms, the completion time of the 2nd process is 14 ms, and the completion time of the 3rd process is determined as 5 ms, and the quantum time is determined as 3 ms, the first process 1 is started. When the 3 millisecond period is over, the first process is temporarily stopped. Then the second process is started and processed for 3 ms again. When the 3 second period is over, the second process is also temporarily stopped and the third process is started. The third process is processed for 3 ms. When the 3 millisecond period is over, the third process is also temporarily stopped and the first process is started by going back to the beginning. The first process continues where it left off. The process continues in this way. Thus, the first, second and third processes are completed and the processes do not wait for each other to finish. It is an intermittent algorithm. The important point here is that the quantum number is well determined, thus the system can be more efficient.

2.4-Particle Swarm Optimization (PSO) - Particle Swarm Optimization

Many tools, ideas and theories used today have been created or imitated by observing nature, animals and people. This is scientifically called biomimicry. Many methods have been introduced to solve information problems inspired by biological systems. For example, artificial neural networks are a simplified model of the human brain. The genetic algorithm is derived from human evolution. Another type of biological system, the social system, examines the interactions of individuals with their environment and other individuals, and their common behaviors. These behaviors are called herd mentality.

The concept of PSO is essentially a simplified simulation of social life. Later, this simulation began to be used as an optimization method. Particle Swarm Optimization (PSO); It is a population-based stochastic optimization technique inspired by the behavior of bird flocks by J. Kennedy and R.C. Eberhart in 1995 [1]. It is designed to solve nonlinear problems. It is used to find solutions to multi-parameter and multivariate optimization problems. PSO has many similarities with evolutionary computing techniques such as genetic algorithms. The system starts with a population of random solutions and searches for the most appropriate solution by updating generations. In PSO, the possible solutions, called particles, follow the optimum particle at that moment and move around in the problem space. Since the number of parameters to be adjusted in the PSO Algorithm is small, its implementation is quite simple. PSO can be successfully applied in many areas such as optimization functions, fuzzy system controls and artificial neural network training [12, 13, 14, 15]. When birds search for food, they follow the bird closest to the food. Each individual solution, called a particle, is like a bird in the search space. When a particle moves, it sends its coordinates to a function to measure the particle's fitness value. A particle must remember its coordinates, speed, the best fitness value it has ever obtained and the coordinates from which it received this value. How its speed and direction in each dimension in the solution space will change each time is determined as a combination of the best coordinates of its neighbors and its own personal best coordinates.The algorithm basically consists of the following steps;

1. Initial swarm is created with randomly generated initial positions and velocities.
2. Fitness values of all particles in the swarm are calculated.
3. Local best (pbest) is found for each particle from the current generation. The number of bests in the swarm is equal to the number of particles.
4. Global best (gbest) is selected from the local bests in the current generation.
5. Positions and velocities are renewed as follows.

Vid=W* Vid + c1*rand1 *(Pid - Xid)+c2*rand2 * (Pid –Xid)

Xid = Xid + Vid

Here Xid gives the position and Vid gives the speed values, while rand1 and rand2 values are randomly generated numbers. W is the inertia weight value and C1, C2 are the scaling factors.

6. Steps 2,3,4,5 are repeated until the stopping criterion is met. [16]

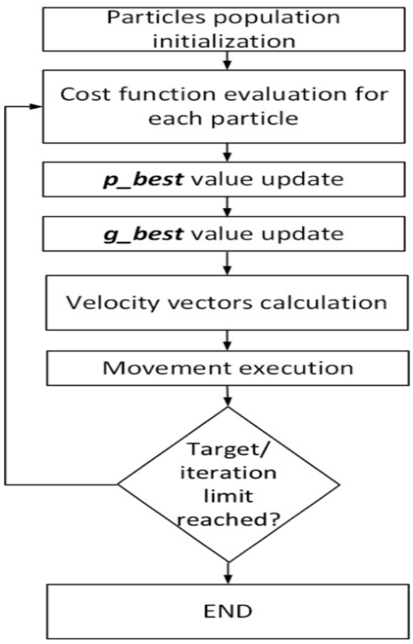

fig 2.2[27]

2.5-Grey Wolf Optimization (PSO)-Grey Wolf Optimization

Volume: 16 Issue: 05 Pages: 6544-6550 (2025) ISSN: 0975-0290

The Grey Wolf Optimization (GWO) algorithm is a population-based metaheuristic optimization algorithm inspired by the social leadership and hunting behaviors of wolves. It was developed by Mirjalili in 2014. Wolves in the pack distribute tasks and take consistent steps when hunting. During the hunt, some wolves are assigned as search wolves and when they find their prey, they report the location of the prey to other wolves by howling. Other wolves approach the prey and surround it. Wolves have a hierarchical structure in four groups within the pack. (See Figure 2.3)

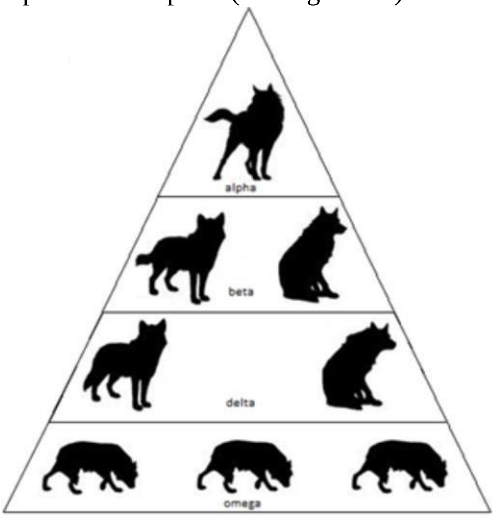

Fig 2.3. Hierarchy of Gray Wolves[28]

Alpha Wolves (α): Leader wolves are called alpha wolves. The leader or alpha wolf is the best wolf in terms of managing other wolves in the group and is usually responsible for making decisions on hunting, sleeping places, waking times, etc. The alpha manages the pack.

Beta Wolves (β): The second wolf in the social hierarchy of the wolf group is the beta wolf. Beta is the assistant of the leader wolf (alpha) in many activities. Betas help in decision-making processes.

Delta Wolves (δ): The delta wolf is the third wolf that must obey the alpha and beta wolves and can only dominate the omega wolves. Hunters and guards fall into the Delta Wolves category. Guards are responsible for monitoring the territory boundaries and warning the colony in case of any danger.

Omega Wolves (ω): The lowest-ranking gray wolf is the omega. Omega wolves must always submit to other dominant wolves.

When expressing it as an algorithm;

The best solution is called alpha (α), the second and third best solutions are called beta (β) and delta (δ), respectively. The remaining candidate solutions are called omega (ω). The steps that make hunting effective and cause it to turn into an Algorithm are as follows.

2.5.1. Surrounding the Prey Step:

During optimization, wolves update their positions around α, β or δ. Gray wolves are formulas created to surround the prey according to equations 3.1 and 3.2.

$$D = |C \cdot XP\ (t) - X(t)| \quad (3.1)$$

$$X(t+1) = X_P\ (t) - A \cdot \mathrm{D} \quad (3.2)$$

Here, X(t) is the position of the wolf and t is the cycle number and XP is the prey position. A and C are the prey position vectors, and X is the position of a gray wolf. The values of A and C are calculated according to equations 2.3 and 2.4.

A=|2.a.r1-a| (3.3.)

C=|2.a.r2 | (3.4)

The components of a lie linearly between 2 and 0 during the iterations, and [0,1] is a random vector between which the random vectors allow the wolves to reach any point in the search space. Thus, the gray wolf can organize its position in the space around the prey at any random location according to equations 3.5 and 3.6. Similarly, the 2- and 3-dimensional space can be expanded to an n-dimensional search space, allowing the gray wolves to move around the best solution obtained so far.

D=|C.Xp(t)-X(t)| (3.5)

X(t+1)=|Xp(t)-A.D| (3.6)

2.5.2. Hunting:

The alpha, beta and delta species of gray wolves have extraordinary knowledge about the current location of the prey. Therefore, the first three best solutions obtained are recorded and other wolves are allowed to update their positions according to the positions of the best search agents. In this context, the following equations can be used.

Dα =|C1.Xα -X| (3.7)

Dβ=|C2.Xβ-X| (3.8)

Dδ=|C3.Xδ-X| (3.9)

X1=|Xα -A1.Dα | (3.10)

X2=|Xβ-A2.Dβ | (3.11)

X3=|Xδ-A3.Dδ | (3.12)

X(t+1)=((X1+X2+X3))/3 (3.13)

Here Dα, Dβ, Dδ are the distances between prey and wolf (alpha, beta, delta); Xα, Xβ, Xδ are the position of prey for alpha, beta and delta wolves; X is the position of the gray wolf in the iteration; C1, C2, C3, A1, A2, A3 are the coefficient vectors for alpha, beta and delta wolves; X1, X2, X3 are the trial vectors for alpha, beta and delta wolves. Figure 2.4. The hunting mechanism of the gray wolf group is shown. In each iteration, the best three wolves are constantly updated. Here, X(t+1) represents the new position of the prey.

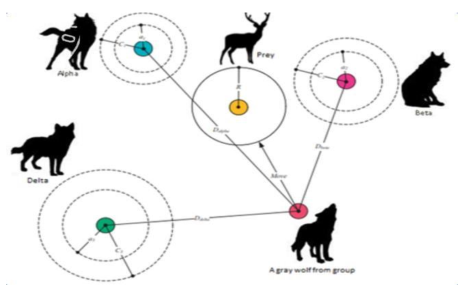

Fig 2.4. Hunting Mechanism of Gray Wolves During Hunting [28]

2.5.3. Attacking the Prey:

Volume: 16 Issue: 05 Pages: 6544-6550 (2025) ISSN: 0975-0290

The approximate distance between the current solution and alpha, beta and delta is calculated. After determining the distances, the final position of the current solution is calculated. At this stage, the value of a is reduced and therefore the range of variation of A is reduced. When A has random values in the range [-1,1], the next position of the search agent will be anywhere between its current position and the position of the prey.

3. Findings

As seen in the graph below (Figure 3.7), we see the process performances of five algorithms with 30 inputs on CloudSim. When categorized, Particle Swarm Optimization (PSO) and Grey Wolf Optimization (GWO) are meta-heuristic algorithms and provided the best performance. Grey Wolf Optimization (GWO) is a more up-to-date algorithm and has shown better performance and superiority over the PSO algorithm. Round Robin, First Come First Serve, and Shortest Job First Algorithms are traditional algorithms and among themselves, First Come First Serve, and Shortest Job First algorithms work continuously, Round Robin algorithm works intermittently and Round Robin algorithm has the worst performance. The time data of the algorithms are shown in the table in Figure 3.9, and the average time of the process performance with 30 inputs on CloudSim is shown in the graph in Figure 3.8.

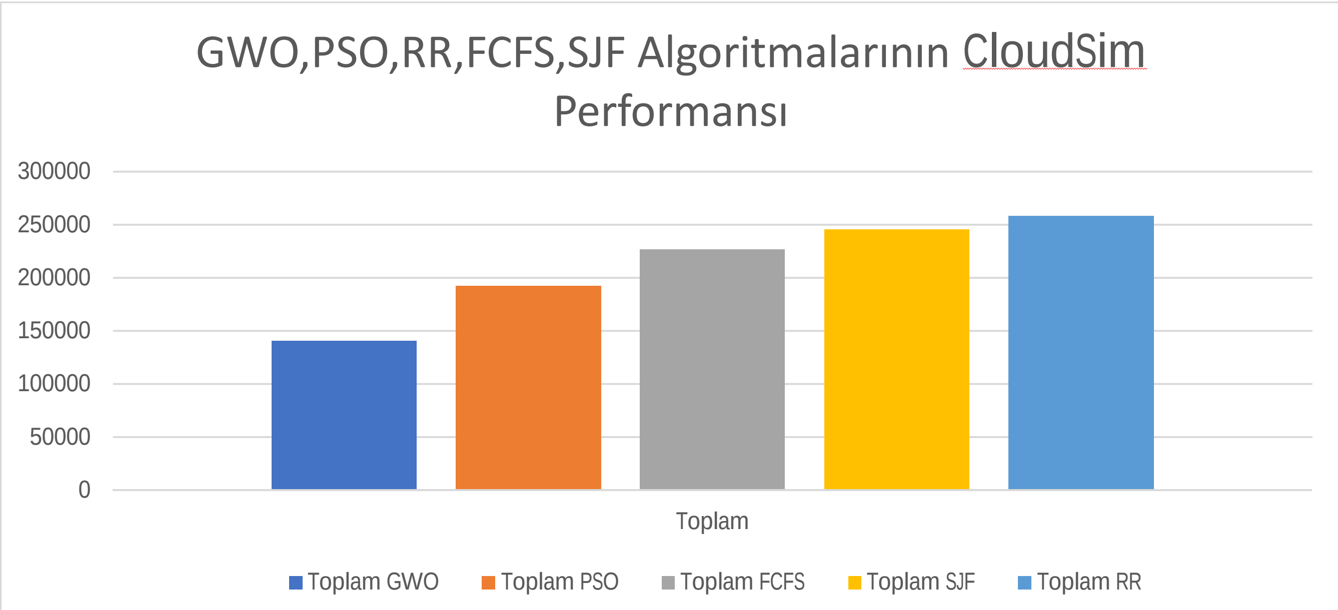

Fig. 3.7 Process Performance Average of All Algorithms.

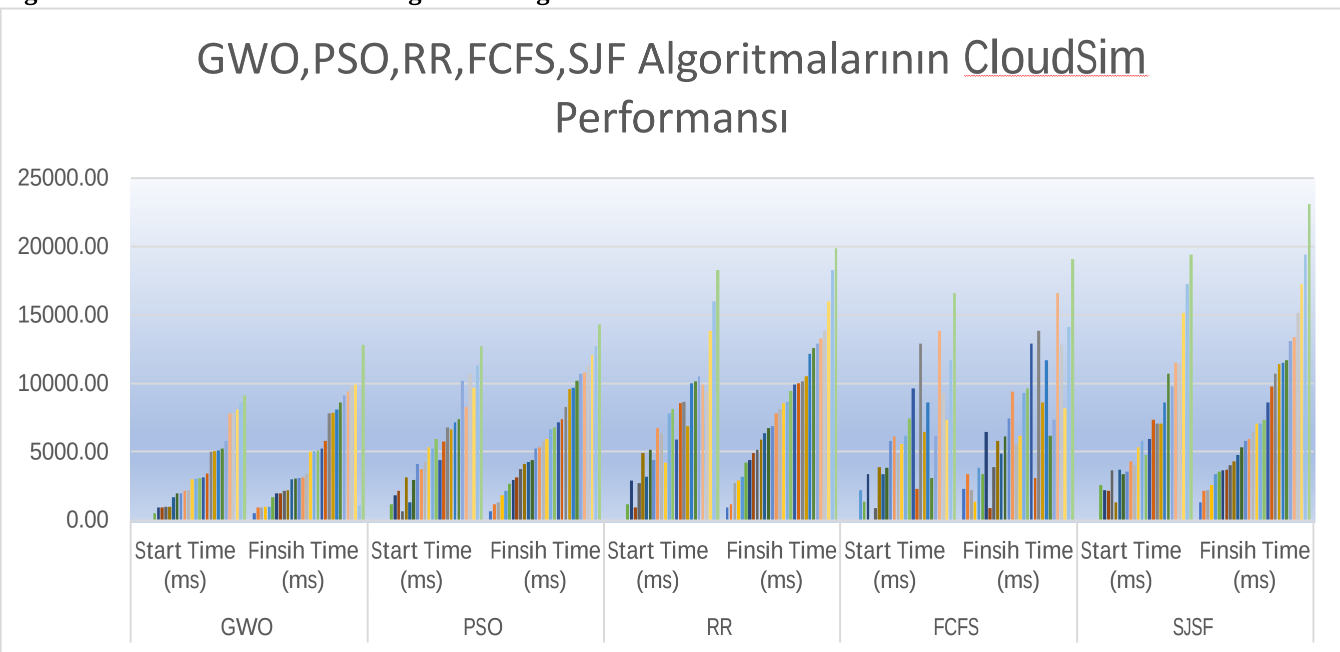

Fig.3.8 Process Performance of All Algorithms at the Start and End

GWO		PSO		RR		FCFS		SJSF	
Start Time (ms)	Finish Time (ms)	Start Time (ms)	Finish Time (ms)	Start Time (ms)	Finish Time (ms)	Start Time (ms)	Finish Time (ms)	Start Time (ms)	Finish Time (ms)
0,20	525,25	0,10	654,07	0,10	973,17	0,20	2315,67	0,10	1307,99
0,20	962,95	0,10	1207,16	0,10	1168,10	0,20	3363,63	0,10	2150,41
0,20	972,29	0,10	1307,99	0,10	2748,74	0,20	2191,80	0,10	2233,50
0,20	973,64	0,10	1827,72	0,10	2929,42	0,20	1393,29	0,10	2589,59
0,20	985,59	0,10	2147,86	0,10	3199,00	2191,80	3841,00	0,10	3390,68
525,25	1700,86	1207,16	2662,50	1168,10	4213,29	1393,29	3402,86	2589,59	3554,99
962,95	1987,84	1827,72	2945,98	2929,42	4410,27	3363,63	6458,91	2233,50	3652,40
972,29	1999,32	2147,86	3138,78	973,17	4920,08	0,20	881,20	2150,41	3735,12
973,64	2145,60	654,07	3773,34	2748,74	5181,34	881,20	3913,45	3652,40	4053,84
985,59	2224,70	3138,78	4121,64	4920,08	5902,94	3913,45	5813,84	1307,99	4313,54
1700,86	3015,92	1307,99	4287,26	3199,00	6377,72	3402,86	4887,72	3735,12	4792,70
1987,84	3073,89	2945,98	4426,83	5181,34	6766,06	3841,00	6124,32	3390,68	5345,68
1999,32	3098,71	4121,64	5264,48	4410,27	6917,63	5813,84	7459,27	3554,99	5798,03
2145,60	3152,07	3773,34	5337,59	6766,06	7823,63	6124,32	9411,24	4313,54	5965,19
2224,70	3452,89	4287,26	5781,94	6377,72	8164,19	4887,72	5566,78	4053,84	6452,89
3015,92	5012,11	5337,59	5942,62	4213,29	8562,43	5566,78	6214,35	5345,68	7063,38
3073,89	5085,55	5264,48	6642,25	7823,63	8660,08	6214,35	9345,36	5798,03	7093,40
3098,71	5120,86	5942,62	6779,08	8164,19	9472,08	7459,27	9652,03	4792,70	7351,06
3152,07	5250,55	4426,83	7194,19	5902,94	9952,74	9652,03	12953,46	5965,19	8607,92
3452,89	5800,11	5781,94	7433,60	8562,43	10017,78	2315,67	3100,32	7351,06	9780,38
5012,11	7810,80	6779,08	8307,33	8660,08	10188,33	12953,46	13848,63	7063,38	10713,90
5085,55	7872,67	6642,25	9611,82	6917,63	10537,28	6458,91	8629,40	7093,40	11442,54
5120,86	8117,49	7194,19	9701,55	10017,78	12187,18	8629,40	11704,63	8607,92	11541,56
5250,55	8617,87	7433,60	10237,21	10188,33	12617,65	3100,32	6190,26	10713,90	11696,76
5800,11	9142,98	10237,21	10743,08	10537,28	12936,33	6190,26	7367,75	9780,38	13102,76
7810,80	9422,22	8307,33	10809,36	9952,74	13319,60	13848,63	16622,12	11541,56	13383,04
7872,67	9823,79	10743,08	11363,18	9472,08	13852,28	9345,36	12849,95	11442,54	15164,54
8117,49	9925,43	9701,55	12093,48	13852,28	16037,96	7367,75	8221,94	15164,54	17273,76
8617,87	1078,18	11363,18	12751,08	16037,96	18308,06	11704,63	14169,00	17273,76	19443,16
9142,98	12828,43	12751,08	14339,05	18308,06	19896,03	16622,12	19122,36	19443,16	23144,87

4. CONCLUSION

As a result, metaheuristic algorithms have proven their superiority by outperforming traditional algorithms. Future studies can be conducted on hybrid systems. Since good performances have been observed from Particle Swarm Optimization (PSO) and Gray Wolf Optimization (GWO) algorithms, to further improve them, hybrid algorithms can be developed by synthesizing them with metaheuristic algorithms having the same orientations and thus better results can be obtained.

References

[1]. M. Aiello, “International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises”, Jan. 2009.

[2]. B. R. Kanani and B. J. Maniyar, “Review on Max-Min Task scheduling Algorithm for Cloud Computing”, vol. 2, no. 3, Mar. 2015.

[3]. K. M. Giannoutakis, C. K. Filelis-Papadopoulos, G. A. Gravvanisand D. Tzovaras, “Evaluation of self-organizing and self-managing heterogeneous high performance computing clouds through discrete-time simulation”, vol. 33, no. 17, Apr. 2021, doi: 10.1002/CPE.6326.

[4]. N. Soltani, B. Soleimaniand B. Barekatain, “Heuristic Algorithms for Task Scheduling in Cloud Computing: A Survey”, vol. 9, no. 8, Aug. 2017, doi: 10.5815/IJCNIS.2017.08.03.

[5]. F. Alhaidari, T. Balharith, and A.-Y. Eyman, ‘Comparative analysis for task scheduling algorithms on cloud computing’, in 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, 2019.

[6]. X. Li, M. Zheng, X. Ren, X. Liu, P. Zhang, and C. Lou, "An Improved Task Scheduling Algorithm Based on Potential Games in Cloud Computing," Lecture notes in computer science, pp. 346–355, Jan. 2014, doi: https://doi.org/10.1007/978-3-319-09265-2_35.

[7]. M. A. Tawfeek, A. El-Sisi, A. E. Keshk, and F. A. Torkey, "Cloud task scheduling based on ant colony optimization," IEEE Xplore, Nov. 01, 2013. https://ieeexplore.ieee.org/document/6707172 (accessed Dec. 24, 2021).

[8]. M. Masdari, F. Salehi, M. Jalali, and M. Bidaki, "A Survey of PSO-Based Scheduling Algorithms in Cloud Computing," Journal of Network and Systems Management, vol. 25, no. 1, pp. 122–158, May 2016, doi: https://doi.org/10.1007/s10922-016-9385-9.

[9]. X. Wang and M. Chen, "Cluster-level feedback power control for performance optimization," Oct. 2008, doi: https://doi.org/10.1109/hpca.2008.4658631.

[10]. C. Li, Amer Qouneh, and T. Li, "Characterizing and analyzing renewable energy driven data centers," Jun. 2011, doi: https://doi.org/10.1145/1993744.1993791.

[11]. H. Mora, S. E. Abdullahi, and S. B. Junaidu, "Modified Median Round Robin Algorithm (MMRRA)," Nov. 2017, doi: https://doi.org/10.1109/icecco.2017.8333325.

[12]. Fabio Alessandro Guerra and Leandro, "Multi-step ahead nonlinear identification of Lorenz's chaotic system using radial basis neural network with learning by clustering and particle swarm optimization," Chaos Solitons & Fractals, vol. 35, no. 5, pp. 967–979, Mar. 2008, doi: https://doi.org/10.1016/j.chaos.2006.05.077.

[13]. J. Zhou, Z. Zhang, Y. Li, J. Deng, and D. Yu, "PSO-based neural network optimization and its utilization in a boring machine," Journal of Materials Processing Technology, vol. 178, no. 1–3, pp. 19–23, Sep. 2006, doi: https://doi.org/10.1016/j.jmatprotec.2005.07.002.

[14]. C. D. Bocaniala and J. Sa da Costa, "Application of a novel fuzzy classifier to fault detection and isolation of the DAMADICS benchmark problem," Control Engineering Practice, vol. 14, no. 6, pp. 653–669, Jun. 2006, doi: https://doi.org/10.1016/j.conengprac.2005.06.008.

[15]. S. P. Ghoshal, "Optimizations of PID gains by particle swarm optimizations in fuzzy based automatic generation control," Electric Power Systems Research, vol. 72, no. 3, pp. 203–212, Dec. 2004, doi: https://doi.org/10.1016/j.epsr.2004.04.004.

[16]. M. Y. Özsağlam and M. Çunkaş, "Optimizasyon Problemlerinin Çözümü için Parçaçık Sürü Optimizasyonu Algoritması", Politeknik Dergisi, vol. 11, no. 4, pp. 299–305, 2008.

[17]. Y. Hou, H. Gao, Z. Wang, and C. Du, "Improved Grey Wolf Optimization Algorithm and Application," Sensors, vol. 22, no. 10, p. 3810, May 2022, doi: https://doi.org/10.3390/s22103810.

[18]. T. Goyal, A. Singh, and A. Agrawal, "Cloudsim: simulator for cloud computing infrastructure and modeling," Procedia Engineering, vol. 38, pp. 3566–3572, 2012, doi: https://doi.org/10.1016/j.proeng.2012.06.412.

[19]. F. Alhaidari and T. Z. Balharith, "Enhanced Round-Robin Algorithm in the Cloud Computing Environment for Optimal Task Scheduling," Computers, vol. 10, no. 5, p. 63, May 2021, doi: https://doi.org/10.3390/computers10050063.

[20]. M. I. Alghamdi, "Optimization of Load Balancing and Task Scheduling in Cloud Computing Environments Using Artificial Neural Networks-Based Binary Particle Swarm Optimization (BPSO)," Sustainability, vol. 14, no. 19, p. 11982, Sep. 2022, doi: https://doi.org/10.3390/su141911982.

[21]. P. K. Gaidhane and M. J. Nigam, "A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems," vol. 27, pp. 284–302, Jul. 2018, doi: https://doi.org/10.1016/j.jocs.2018.06.008.

[22]. X. Huang, C. Li, H. Chen, and D. An, "Task scheduling in cloud computing using particle swarm optimization with time varying inertia weight strategies," Cluster Computing, vol. 23, no. 2, pp. 1137–1147, Sep. 2019, doi: https://doi.org/10.1007/s10586-019-02983-5.

[23]. Abed, A. H. (2025). The Techniques of authentication in the Context of Cloud Computing. Int. J. Advanced Networking and Applications, 16(04), 6515-6522. https://doi.org/10.35444/IJANA.2025.16408

[24]. AlRababah, A. A. (2025). Integrating Computing Techniques for Parametric Sensitivity Analysis in Control System Optimization. Int. J. Advanced Networking and Applications, 16(04), 6453-6459. https://doi.org/10.35444/IJANA.2025.16401

[25]. Khan, Mohd Sha Alam, and R. Santhosh. "Task scheduling in cloud computing using hybrid optimization algorithm." Soft computing 26.23 (2022): 13069-13079.

[26]. Anonim, 2021 https://www.geeksforgeeks.org/what-is-cloudsim/

[27]. Koç, Tarık, İbrahim Eke, and Suleyman Sungur Tezcan. "Parçacık Sürü Optimizasyonu ve Genetik Algoritma Kullanılarak Birleşik Isı ve Güç Ekonomik Dağıtımı Probleminin Çözümü." International Journal of Engineering Research and Development 13.3 (2021): 230-241.

[28]. İnaç, Tufan, Emrah Dokur, and Uğur Yüzgeç. "A multi-strategy random weighted gray wolf optimizer-based multi-layer perceptron model for short-term wind speed forecasting." Neural Computing and Applications 34.17 (2022): 14627-14657.