A Comparative Analysis for Energy Efficiency in Cloud Computing using CSO

Sabyasachi Narendrasingh
M.Tech Scholar
Department of CSE
OUTR, Bhubaneswar

Ashis Kumar Mishra
Department of CSE
OUTR, Bhubaneswar

Subasish Mohapatra, PhD
Department of CSE
OUTR, Bhubaneswar

ABSTRACT
Now a day, many organizations and independent consumers are heavily utilizing cloud services for their scalability, reliability, and low expenses. Developing management techniques used to minimize energy consumption, boost profitability, and decrease environmental impact is a key aspect of cloud data centers. Every consumer's usage of service can generate a huge amount of data. Therefore, it will be very expensive to transfer data between two dependent resources. This paper suggests a meta heuristic-based optimization process called the Cat swarm optimization algorithm to improve the performance and energy efficiency in cloud resource allocation. Also, its performance is compared with gray wolf optimization and Whale Optimization Algorithms with a variation in of population sizes and iterations. Also, energy efficiency and throughput are calculated.

General Terms
Comparative analysis for CSO, GWO and WOA.

Keywords
Datacenters, CSO, GWO, WOA, Cloud Computing, Energy Efficiency, Throughput.

1. INTRODUCTION
Cloud computing offers secure and comfortable services. Many individuals and organizations depend on it to be reliable as well as cost-effective. [1-3]. Cloud computing is a primary method for lessening computing costs for customers. The aid of this method, the creation and generation used for further processing, generating, and managing IT in the organization, as well as the supplier, are inspired. Cloud computing uses physical and digital equipment, virtual, the allocation of computing resources, and customers. Customers demand to access various computing resources along with memory via the internet from anywhere and as long as they need. The service level agreement is primarily used by the cloud to deliver services and resources to the client. Users can access services through the cloud to improve their performance and reduce costs. A lot of electricity is required by the data centers that host the cloud services. Energy consumption needs to be reduced inside cloud data centers in order boost the revenue for cloud distributor while decreasing consumer costs and environmental impact [4-5]. Cloud service hosting data centers require a lot of power. Energy use inside cloud data centers must be reduced in order to increase revenue for the cloud distributor, lower costs for clients, and reduce environmental impact [2-3]. Cloud computing, in contrast to grid or utility computing. In facts, it is an autonomous computing platform. Grid computing and utility computing are not the same as cloud computing. It is a fairly independent platform in computing.

1.1 Load Balancing
A technique used in cloud computing is load balancing for evenly allocating workloads among various computational resources, including servers and virtual machines. To make the distributed system run quicker and more efficiently, distribute the load to each node or computer Load balancing is used to decrease response times, increase resource efficiency, and overload specific resources. The total amount of computing time performed by a system is measured by its load. CPU load, memory use, and network delay load are the many forms of load.

2. ENERGY EFFICIENCY
Energy efficiency in cloud computing is the process to make the best use of energy resources to reduce energy usage and data center carbon footprint. Utilizing shared computing resources is known as cloud computing supplied through the internet, such as apps, servers, and storage. Data centers consumed a large amount of power due to cloud workloads High-power consumption results in a significant amount of heat production [5]. It may affect hardware dependability and increase the burden on cooling systems. As a result, large operational expenses in the form of increasing electricity bills. Energy consumption must be reduced inside the expansion of the cloud data centers revenue for cloud distributors while decreasing consumer costs and environmental impact.

2.1 Energy consumption in cloud
The CPU, RAM, and disc storage are used to calculate energy usage in cloud data centers. A Linear model is used as a general energy consumption model in the cloud. Increasing the revenue on behalf of cloud provider sources and dropping the lower cloud cost for all users affect the environment. There are some advantages of energy optimization in cloud data centers. The energy performance of the cloud is using data storage, magnetic disc, and user-shared Virtual Machines. There are several more comprehensive methods that are used for the management of energy in the cloud. The primary goals of the processes applied to cloud energy management are explained below.

1) To apply all resources to various clients effectively and efficiently, one of the most famous techniques called optimization methods is applied. The optimization approach is used for assessing resource accessibility for the optimal distribution of different resources.

2) Energy usage and consumption are tied to network aspect inspections, including CPU, RAM, and disc availability. The objective is to provide some sort of solution for sharing resources based on purpose rather than comparable forms of Workloads

2.2 Optimization Method for Energy Management

For a number of cloud-based data centers, some researchers used the optimization technique. Here are all the specifics used to optimize energy consumption in a cloud environment applying optimization approaches are given in this fig 1.

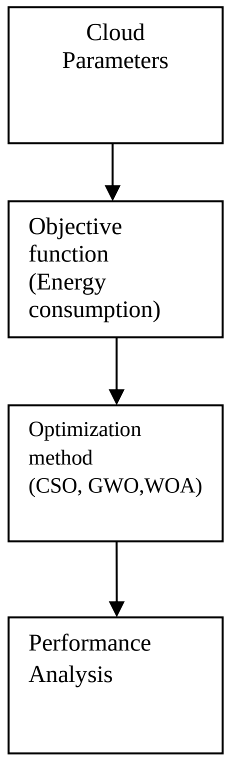

Fig 1: Energy optimization overall model

This strategy has an impact on how the business and the service provider process build and manage IT. The four main components of cloud computing are resource allocators, clients, virtual machines, and physical machines. Users required access to cloud resources such as CPU, RAM, and physical memory. The cloud offers all the resources and services they need to carry out their user tasks according to the SLA [4,5]. To deliver services to users, virtual machines are distributed in cloud-based PMs. The estimate required to provide the specified workloads is required by VMs. Therefore, energy efficiency must be reduced in cloud data centers in order to increase cloud service provider(CSP) profits, lower user costs, and reduce environmental impact[7,8]. There is a requirement for an aback-and-forth explanation for the low energy utilization of minimum energy and optimized throughput. Optimization refers to the process of finding the best possible solution from a set of alternatives for a given issue.

Researchers have developed many algorithms based on swarm intelligence for achieving balance and deliver improved responses to optimization challenges. Chu et al. first developed the swarm Intelligence algorithm known as Cat Swarm Optimization. 2006[14].this algorithm has a novel approach to designing exploration and exploitation phases and it is based on the cats' natural behavior. This novel algorithm has been successfully used in several scientific and engineering optimization fields. While many new algorithms were invented since then, it has been compared to the PSO algorithm most of the time

3. LITERATURE SURVEY

N. Prabhakaran et al. has been proposed a novel OCSODL-CCFD approach for detecting and classifying fraudulent credit card transactions. The OCSO algorithm's architecture aids in reducing computing complexity and improving classification results [6].

Xiang Li et al. [7] has been suggested a model for a collaborative low-carbon power grid technology that is based on the ICSO algorithm. Approach to produce a more stable and environmentally friendly power grid operation state while maintaining a minimal construction cost.

Xiao-Fang Ji et al. [8] has been proposed a model innovative approach named as Adaptive Cat Swarm Optimization. It provides improved search results by combining the advantages of two swarm intelligence algorithms, CSO and APSO. When it comes to exploration and exploitation, ACSO outperforms all other current heuristics. However relevant parameter changes take much longer processing time.

Yadav et al. [9] have applied the bandwidth selection strategy (Bw), the gradient descent-based regression, and the maximize correlation percentage for energy management and SLA violation. As a result, the energy consumption and SLA violation were reduced by performing VM selection and overloading the host. According to SLAs, the VM is chosen using bw policy based on the host's network traffic. As a result, the performance was improved, energy consumption decreased, and the developed method's computational time was deemed acceptable for the analysis. However, the model learning rate was low, making it susceptible to local optima.

Xie et al. [10] have proposed a model for a cyber-physical system, the global energy savings for multiple workloads (GESMW) concept. Here, the GESMW approach is applied to achieve the energy consumption deadline. In this investigation, the various types of processes were typically examined for efficient management. Then it was discovered that the GESMW model had demonstrated improved vitality management and had met the workload deadline. However, the model did not examine the CPS's physical environment in relation to the energy management strategy.

Yi Cai et al. [11] have proposed a method Using the Tabu genetic cat swarm optimization technique, mistuned blades can be optimized. The parallel algorithm is both inexpensive and efficient. The optimization scheme noticeably reduces the bladed disc system's amplitudes of vibration.

Karthikeyan et al. [12] have proposed to decrease Utilizing an artificial bee colony and bat hybrid optimized techniques and Naive Bayes to help cut down on energy use when moving virtual machines. The approach was carried out with Cloud, which analyzed the suggested algorithm's energy use, failure, and success rates. The optimization strategy improved VM migration performance significantly. And the model demonstrated improved energy management effectiveness and success rates. This method's weaknesses included lower convergence as well as getting easily trapped in local optima.

The overall game theory task arranging algorithm was suggested by Yang et al. [13]. This approach is used for energy administration. For dependability, the balanced arranging technique was utilized, while the assigned arranging model was used for processing nodes. The stability condition was applied to the objective of the function. In terms of power management, the overall game theory method provided significantly more efficient performance correlated with optimization. Power consumption was lowered while QoS was improved significantly. This analysis fails to include the price of processing of the specific network, and also the fewer prices necessary for power management.

A.Seyyedabbasi et al.[14] has been applied SCSO algorithm to evaluate using 20 well-known and current test functions from the CEC2019 benchmark, and the findings are analyzed by using well-known metaheuristic algorithms. SCSO achieved the highest performance in 63.3% of the test functionalities. The purposed algorithm has been used in the fixing of seven challenging engineering design issues.

Li et al. [15] have been purposed a tradeoff strategy that is used in a special way for optimum strength usage and waiting time within a cloud. For several levels, the wait threshold and even energy use have been examined. Even energy consumption has been examined for various layers, including the wait threshold. To reduce the amount of power used inside the unit, there was a cloud-fog scheduling technique to be suggested in various cloud layers, the developed approach produced higher energy consumption rates. Typically, the technique for VM migration appeared inefficient, and even a developing optimization method was required to give an efficient solution.

The cat swarm optimizations and the firefly algorithm have been integrated into an integrated multi-objective scheduling approach by Du, Y.an et al. [16]. using swarm intelligence methods to fix problems with job-shop scheduling. The CSO-FA has been used to shorten the travel time to the global optimum. The developed model was demonstrated to be effective at handling multi-objective scheduling of cloud-based manufacturing services.

Ricardo Soto et al. [17] have been proposed a novel Binary Cat Swarm Optimization for solving the Manufacturing Cell Design Problem. The main objective was to find a cell structure that reduces the movement of various components between cells. It shows the BCSO is a feasible different to solving the MCDP, and that was might be enhanced by applying various parameters for each group case.

Fernández- Cerero et al. [18] has be proposed SCORE tool. This type of research created a cloud optimization simulator for their usage resources. The concurrent scheduling Approach was created to reduce the amount of energy consumed by actual, and simulated workloads. When it comes to cloud scheduling and overall energy efficiency, the developed model performed better. And with this approach, the workload and QoS constraints were not handled seriously.

3.1 Observation

The observations from the literature are as follows;

I. The increase in power consumption has become an important issue in a cloud environment.

II. Existing optimization algorithm generates the highest computational cost and is less efficient.

III. For varying the cluster size, the existing optimization method is not sufficient to produce an effective result.

IV. Existing optimization techniques have weaknesses including lower convergence as well as getting easily trapped in local optima.

V. Power consumption was lowered while QoS was improved significantly.

3.2 Motivation

Energy efficiency must be reduced in cloud data centers in order to increase cloud service providers, profits, lower user costs, and reduce environmental impact. Since it was first introduced, CSO has been praised because it is a powerful and successful metaheuristic swarm-based optimization technique. This algorithm has a novel approach to designing phases of exploration and exploitation and it is based on the cats' natural behavior. The CSO algorithm, like any other metaheuristic algorithm, has benefits and drawbacks. The tracing mode and seeking modes are separated and independent.

While many new algorithms were invented since then, it has been compared to the PSO algorithm most of the time. Since the CSO has not been explored yet and according to many researchers GWO and WOA are best suitable for optimization. Therefore, in terms of fitness value, the performance of CSO should be contrasted with that of two optimization strategies. The searching mode is for local searches, while the tracing mode is for global searches. Also, energy efficiency and throughput will be calculated with five different tasks..

4. PROPOSED MODEL

4.1 Cat Swarm Optimization Algorithm

The cat Swarm Optimization Algorithm is a Metaheuristic Optimization Algorithm. It comes in the category of Nature Inspired Swarm Based Optimization Algorithm. As we know Nature Inspired Swarm Based Optimization Algorithms are unpredictable procedures. They are intended to address various optimization issues. The algorithm for Cat Swarm Optimization was inspired by real-life cat behavior. CSO was invented in 2006 by Shu Chuan Chu. The author tested Cat Swarm Optimization Algorithm by using 23 benchmark functions from the classical era and 10 from the current era. It is obvious that how cats behave in the real world served as the inspiration for Cat Swarm Optimization. Cats are most of the time inactive, they have a strong curiosity.

CSO was originally a single-objective, continuous algorithm [13]. The resting and tracking behaviors of cats' served as its inspiration. The Cats seem lazy and spend a lot of their free time doing nothing. Relaxing, but when they do, they are extremely conscious and aware of their surroundings. As a result, they constantly keep an intelligent and deliberate eye on their surroundings, and as soon as they spot something they want to get closer to, that's when they move quickly in that direction. Consequently, the CSO method is founded on the union of two important cat behaviors.

CSO algorithm consists of two modes, such as tracing and seeking modes. i. e. relaxing, looking around, or stuck in a job state to maneuver to another location. In doing a trace for mode Cats are active we. They change their recent position. Each of the cats stands for a response set, which includes personal position, fitness value, and flag [25]. The positioning in the search space is usually made up of M sizes, every dimension has a different velocity, and the fitness value described amply illustrates the best-case scenario. The flag will classify the cats as being either in searching mode or maybe in tracing mode. In order to run the algorithm, we must first decide how many cats will be measured in each iteration. Typically, each iteration's best cat is retained in memory, and the most recent iteration will be used as the outcome.

4.2 Seeking Mode

In this mode, which replicates cats' resting habits and four key factors are important. There is self-position consideration (SPC), SMP stands for seeking memory pool, SRD for seeking a range of the chosen dimension, and CDC for seeking counts of dimension to change. Invoking trial-and-error, the user procedure to adjust the settings as exactly as possible. The extent of a cat's seeking memory is determined by SMP, and how many dimensions there will be updated is define by the CDC and is in the range [0, 1]. For instance, SMP would be set to 5 then 5 new random locations would be produced for each cat, with one of them being chosen to be the cat's next position. Each cat must alter four of the five randomly generated dimensions while maintaining the fifth as constant if the CDC is set to 0.2 and the search space has five dimensions. The mutative ratio of chosen dimensions is known as SRD. The amount of mutation and alteration is specified for the dimensions the CDC has selected. Whenever the SPC flag is true, we must produce candidates rather than using the SMP number since the current positions are considered when determining whether the cat's position is chosen as a candidate position for the following iteration.

The steps for the following are seeking modes:

1. Create as possible SMP copies for the present position of Catk.

2. Choose as many CDC dimensions at random for each copy that will be mutated. Randomly replace the old positions by adding or subtracting from the SRD values, present values. It has laced the old positions shown in equation 1:

$$Xj\ dnew = (1+rand*SRD) * Xj\ dold \quad (1)$$

3. Then calculate the fitness value (FS) for each cat’s position.

4. Define one of the candidate points as the primary point and that point will be the cat's next position, where candidates with higher FS have a greater chance of being chosen, as indicated by the previous equation. However, if all fitness values are equal, make sure that each candidate point's selection probability is set to 1.

$$Pi = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}}, \text{ where } 0<i<j \quad (2)$$

When it is asked for minimization, apply FSb = FSmax ; otherwise, FSb = FSmin

4.3 Tracing Mode

This concept is based on how cats track objects. All a cat's position dimensions are given random velocity values for the initial iteration. We must change the velocity values afterward.

These are the steps in tracing mode:

1. Update the velocities (Vk,d) for all dimensions by using equation (3).

2. When the velocity value is more than the maximum value, then it is equal to the maximum velocity.

$$V_{k,d} = V_{k,d} + r_1 c_1 (X_{best,d} - X_{k,d}) \quad (3)$$

3. Update the position value of Catk using the following equation.

$$X_{k,d} = X_{k,d} + V_{k,d} \quad (4)$$

4.4 Flow chat of proposed Cat Swarm Optimization Algorithm

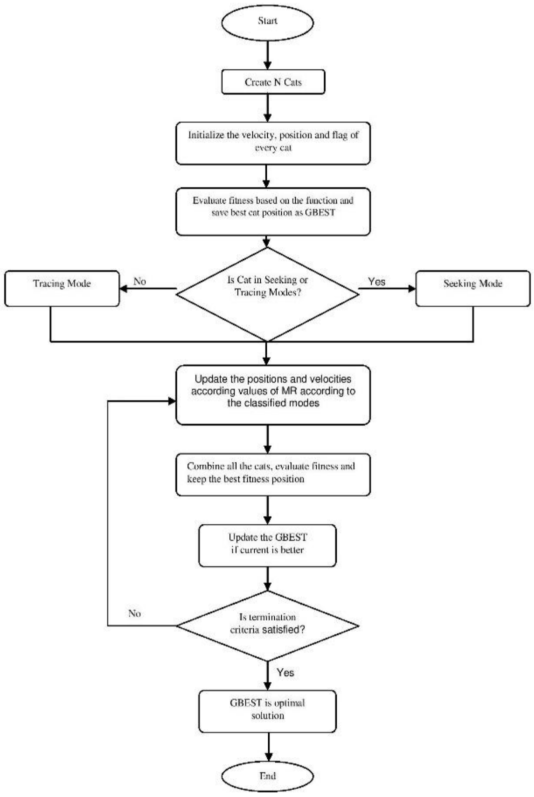

Fig 2: Work flow of CSO

5. EXPERIMENTAL RESULTS ANALYSIS

The result of proposed work are analyzed and discussed. The capability of proposed model was evaluated using python environment. The objective was to find out the best suitable condition and technique for optimization of energy efficiency. This work tried to find out which among three techniques is best suitable with and what is the best value for population size and how many iteration should be best fit. Values taken for number of iterations were 20, 40, 60, 80 and 100. Similarly values taken for population size were 20, 50, and 100. Different values are used as these parameters and the best fitness value is calculated which is shown in table 2, 3, 4 and graphs are shown in figure in 3,4 and 5.

Comparison between the outputs based on different parameters were done and shown in figure 3,4, and 5.When 20 was used as population size, and CSO gave best result at 60 iterations. GWO and WOA performed better at 40 and 60 iterations respectively. When 50 populations were applied, the best result shown was by CSO at 80 and 100 iterations. Likely GWO and WOA performed better at 80 and 100 iterations respectively. When population size was 100, CSO gave best and consistent result at 40, 60, 80 and 100 iterations. Best result generated by GWO was very close to CSO at 60 iterations. Performance of WOA was better when 80 iterations were applied. In general the best result was generated by CSO algorithm when population size is 50 and iterates for 100 times.

After finding the best environment for individual optimization techniques, energy efficiency and throughput are calculated which is shown in table 5. When energy efficiency and throughput is calculated for five different tasks, in case of CSO energy efficiency varies from 2.48 to 15.31 and throughput varies from 12.21 to 1.80. In case of GWO energy efficiency varies from 3.50 to 17 and throughput varies from 11.07 to 5.99. When WOA is applied for those 5 tasks energy efficiency varies from 5.92 to 18.21 and throughput varies from 10.01 to 4.26. Figure 6 shows the graph between energy efficiency and throughput

Table 1. Parameters for CSO, GWO, WOA.

CSO		GWO		WOA	
Parameter	**value**	**Parameter**	**Value**	**Parameter**	**value**
Initial cats	100	No. of wolves	100	No. of agents	100
No. of Iteration	100	No. of Iteration	100	No. of Iteration	100
SMP	5	r1, r2	[0,1]	r1, r2	[0,1]
r	[0,1]	Dimension	5	dimension	5
CDC	0.2				

Table 2. Best fitness value from 20 population size in CSO,GWO and WOA

	20 Iterations	40 Iterations	60 Iterations	80 Iterations	100 Iterations
CSO	28.52368660446535 4	22.10590250833332	12.659353735228834	18.363544396365253	18.363544396365253
GWO	25.28208517237499	33.822894042331406	35.00401399959954	22.469476813577952	28.67555055976588
WOA	32.27538812460874	28.108875864314147	38.29839139037345	22.2340424495393	13.587581018799282

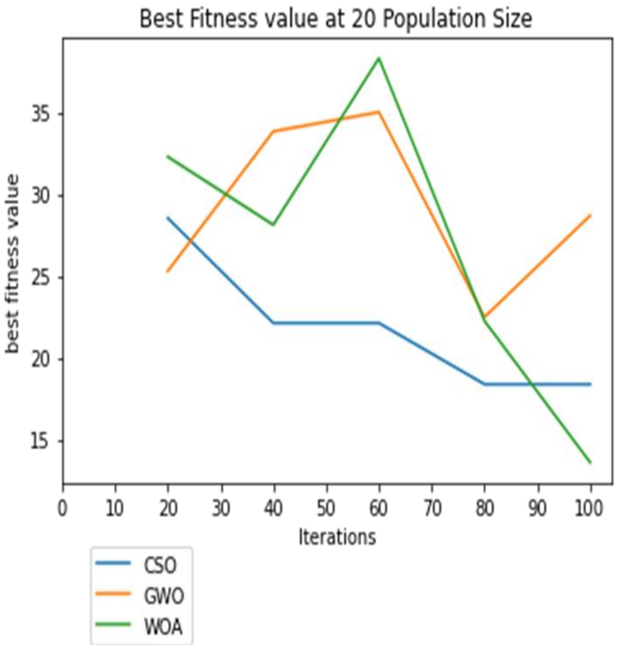

Fig 3: Comparison of best fitness value among the CSO, GWO and WOA at 20 populations.

Table 3. Best fitness value from 50 population size in CSO GWO and WOA

	20 Iterations	**40 Iterations**	**60 Iterations**	**80 Iterations**	**100 Iterations**
CSO	18.6979661822784 92	18.6879662824864 92	15.783240417735222	11.874555094907052	11.8855485084908503
GWO	27.54075409342675	23.57126653017358 8	20.680719987293624	31.34920307270604	22.285120373545546
WOA	22.37544151675294	30.29038678945066	29.420061060686603	26.02917511467644	26.899168700431886

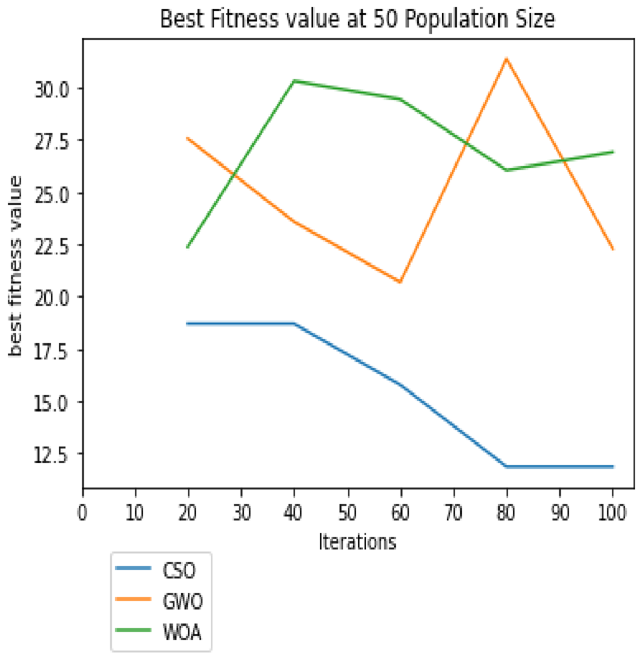

Fig 4: Comparison of best fitness value among the CSO, GWO and WOA at 50 populations.

Table 4. Best fitness value from 100 population size in CSO GWO and WOA

	20 Iterations	40 Iterations	60 Iterations	80 Iterations	100 Iterations
CSO	16.895600428887022	12.759353735228834	12.689353735228534	12.859353734228736	12.659353735229834
GWO	22.224675248173796	33.962092529734605	32.41178689519725	27.416047437939994	26.244653850756773
WOA	22.630571084387416	29.567673994576346	26.77588668981868	15.48957044810848	24.941743134715963

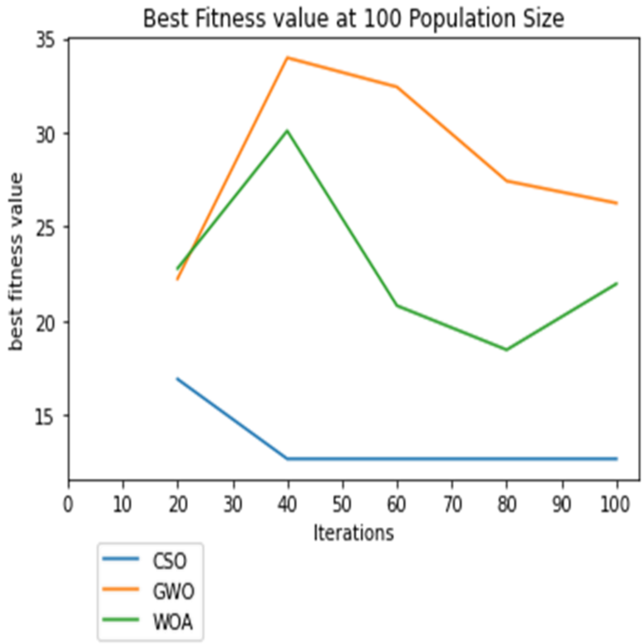

Fig 5: Comparison of best fitness value among the CSO, GWO and WOA at 100 populations.

Table 5. Energy Efficiency and Throughput of different algorithm

CSO		GWO		WOA	
Energy Efficiency	**Throughput**	**Energy Efficiency**	**Throughput**	**Energy Efficiency**	**Throughput**
2.48	13.21	3.50	11.07	5.92	10.01
4.48	9.27	6.08	8.59	7.10	9.03
8.59	8.81	9.00	7.91	10.28	6.46
11.97	3.85	13.69	6.99	13.66	5.00
15.31	1.80	17.00	5.99	18.21	10.01

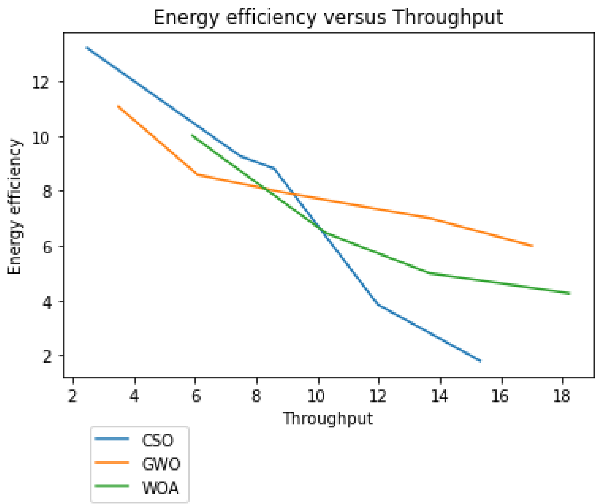

Fig 6: Graph of Energy Efficiency versus Throughput.

6. CONCLUSION AND FUTURE WORK

In this research work three optimization based algorithms are experimentally explored and studied i.e. CSO, GWO and WOA. Here, two parameters are focused: one is number of iterations and second one is population size. The two parameters executed in various condition and values to obtained best optimization result in field of energy consumption. Also energy efficiency and throughput is calculated for five different tasks. It is found that CSO algorithm performs better when population size is 50 and number of iterations is 100 and energy efficiency was very low as compared to GWO and WOA. In this observation several parameters has been examined during experimental study to achieve better performance as well as showing promising results in solving multi-objective optimization and real-world applications. Since this algorithm is not being explored a lot, global optimization is challenging. So, in future the algorithm can be studied and explored to build hybrid models to solve real life computing problems in various scientific domains.

7. REFERENCES

[1] Moganarangan, N., Babukarthik, R. G., Bhuvaneswari, S., Basha, M. S., & Dhavachelvan, P. (2016). A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach. Journal of King Saud University-Computer and Information Sciences, 28(1), 55-67.

[2] Ficco, M., & Palmieri, F. (2015). Introducing fraudulent energy consumption in cloud infrastructures: A new generation of denial-of-service attacks. IEEE Systems Journal, 11(2), 460-470.

[3] Ahvar, E., Orgerie, A. C., & Lebre, A. (2019). Estimating Energy Consumption of Cloud, Fog, and Edge Computing Infrastructures. IEEE Transactions on Sustainable Computing, 7(2), 277-288.

[4] El Kafhali, S., & Salah, K. (2018). Modeling and analysis of performance and energy consumption in cloud data centers. Arabian Journal for Science and Engineering, 43(12), 7789-7802.

[5] Vishwanath, A., Jalali, F., Hinton, K., Alpcan, T., Ayre, R. W., & Tucker, R. S. (2015). Energy consumption comparison of interactive cloud-based and local applications. IEEE Journal on selected areas in communications, 33(4), 616-626.

[6] Prabhakaran, N., & Nedunchelian, R. (2023). Oppositional Cat Swarm Optimization-Based Feature Selection Approach for Credit Card Fraud Detection. Computational Intelligence and Neuroscience, 2023.

[7] Li, X., Guo, C., Li, C., Xu, T., & Wu, S. (2022). Power Grid Low Carbon Collaborative Planning Method Using Improved Cat Swarm Optimization Algorithm in Edge Cloud Computing Environment. Wireless Communications and Mobile Computing, 2022.

[8] Ji, X. F., Pan, J. S., Chu, S. C., Hu, P., Chai, Q. W., & Zhang, P. (2020). Adaptive cat swarm optimization algorithm and its applications in vehicle routing problems. Mathematical Problems in Engineering, 2020, 1-14.

[9] Yadav, R., Zhang, W., Kaiwartya, O., Singh, P. R., Elgendy, I. A., & Tian, Y. C. (2018). Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing. IEEE Access, 6, 55923-55936.

[10] Xie, G., Zeng, G., Jiang, J., Fan, C., Li, R., & Li, K. (2020). Energy management for multiple real-time workflows on cyber–physical cloud systems. Future Generation Computer Systems, 105, 916-931.

[11] . Cai, Y., Gu, J., Pan, H., Zhang, H., & Zhao, T. (2021). Tabu Genetic Cat Swarm Algorithm Analysis of Optimization Arrangement on Mistuned Blades Based on CUDA. Shock and Vibration, 2021, 1-18.

[12] Karthikeyan, K., Sunder, R., Shankar, K., Lakshmanaprabu, S. K., Vijayakumar, V., Elhoseny, M., & Manogaran, G. (2020). Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA). The Journal of Supercomputing, 76, 3374-3390.

[13] Yang, J., Jiang, B., Lv, Z., & Choo, K. K. R. (2020). A task scheduling algorithm considering game theory designed for energy management in cloud computing. Future Generation computer systems, 105, 985-992.

[14] Seyyedabbasi, A., & Kiani, F. (2022). Sand Cat swarm optimization: A nature-inspired algorithm to solve global optimization problems. Engineering with Computers, 1-25.

[15] Li, G., Yan, J., Chen, L., Wu, J., Lin, Q., & Zhang, Y. (2019). Energy consumption optimization with a delay threshold in cloud-fog cooperation computing. IEEE access, 7, 159688-159697.

[16] Du, Y., Wang, J. L., & Lei, L. (2019). Multi-objective scheduling of cloud manufacturing resources through the integration of Cat swarm optimization and Firefly algorithm. Advances in Production Engineering & Management, 14(3).

[17] Soto, R., Crawford, B., Aste Toledo, A., Castro, C., Paredes, F., & Olivares, R. (2019). Solving the manufacturing cell design problem through binary cat swarm optimization with dynamic mixture ratios. Computational intelligence and neuroscience, 2019.

[18] Fernández-Cerero, D., Fernández-Montes, A., Jakóbik, A., Kołodziej, J., & Toro, M. (2018). SCORE: Simulator for cloud optimization of resources and energy consumption. Simulation Modelling Practice and Theory, 82, 160-173.

[19] Pappula, L., & Ghosh, D. (2018). Cat swarm optimization with normal mutation for fast convergence of multimodal functions. Applied soft computing, 66, 473-491.

[20] Nie, X., Wang, W., & Nie, H. (2017). Chaos quantum-behaved cat swarm optimization algorithm and its application in the PV MPPT. Computational Intelligence and Neuroscience, 2017.

[21] Gabi, D., Ismail, A. S., Zainal, A., Zakaria, Z., & Al-Khasawneh, A. (2017, May). Cloud scalable multi-objective task scheduling algorithm for cloud computing using cat swarm optimization and simulated annealing. In 2017 8th International Conference on Information Technology (ICIT) (pp. 599-604). IEEE.

[22] Kumar, Y., & Singh, P. K. (2018). Improved cat swarm optimization algorithm for solving global optimization problems and its application to clustering. Applied Intelligence, 48, 2681-2697.

[23] Naim, A. A., El Bakrawy, L. M., & Ghali, N. I. (2017). A hybrid Cat Optimization and K-median for Solving Community Detection Problem. Asian Journal of Applied Sciences, 5(5).

[24] Hadi, I., & Sabah, M. (2014). Enhanced hybrid cat swarm optimization based on fitness approximation method for efficient motion estimation. International Journal of Hybrid Information Technology, 7(6), 345-364.

[25] Bilgaiyan, S., Sagnika, S., & Das, M. (2014, February). Workflow scheduling in cloud computing environment using cat swarm optimization. In 2014 IEEE International Advance Computing Conference (IACC) (pp. 680-685). IEEE.

[26] . Peng, H., Wen, W. S., Tseng, M. L., & Li, L. L. (2019). Joint optimization method for task scheduling time and energy consumption in mobile cloud computing environment. Applied Soft Computing, 80, 534-545.

[27] Strumberger, I., Bacanin, N., Tuba, M., & Tuba, E. (2019). Resource scheduling in cloud computing based on a hybridized whale optimization algorithm. Applied Sciences, 9(22), 4893.

[28] Chang, Z., Zhou, Z., Ristaniemi, T., & Niu, Z. (2017, December). Energy efficient optimization for computation offloading in fog computing system. In GLOBECOM 2017-2017 IEEE Global Communications Conference (pp. 1-6). IEEE.

[29] Yan, D., Cao, H., Yu, Y., Wang, Y., & Yu, X. (2020). Single-objective/multiobjective cat swarm optimization clustering analysis for data partition. IEEE Transactions on Automation Science and Engineering, 17(3), 1633-1646.

[30] Zhong, W., Zhuang, Y., Sun, J., & Gu, J. (2018). A load prediction model for cloudcomputing using PSO-based weighted wavelet support vector machine. Applied Intelligence, 48, 4072-4083.

[31] Vasudevan, M., Tian, Y. C., Tang, M., Kozan, E., & Zhang, X. (2018). Energy-efficient application assignment in profile-based data center management through a repairing genetic algorithm. Applied Soft Computing, 67, 399-408.

[32] Zekić-Sušac, M., Mitrović, S., & Has, A. (2021). Machine learning based system for managing energy efficiency of public sector as an approach towards smart cities. International journal of information management, 58, 102074.

[33] Askarizade Haghighi, M., Maeen, M., & Haghparast, M. (2019). An energy-efficient dynamic resource management approach based on clustering and meta-heuristic algorithms in cloud computing IaaS platforms: Energy efficient dynamic cloud resource management. Wireless Personal Communications, 104, 1367-1391.

[34] Prassanna, J., & Venkataraman, N. (2021). Adaptive regressive holt–winters workload prediction and firefly optimized lottery scheduling for load balancing in cloud. Wireless Networks, 27, 5597-5615.

[35] Arianyan, E., Taheri, H., & Khoshdel, V. (2017). Novel fuzzy multi objective DVFS-aware consolidation heuristics for energy and SLA efficient resource management in cloud data centers. Journal of Network and Computer Applications, 78, 43-61.

[36] Horri, A., & Dastghaibyfard, G. (2015). A novel cost based model for energy consumption in cloud computing. The Scientific World Journal, 2015.

[37] Hanini, M., Kafhali, S. E., & Salah, K. (2019). Dynamic VM allocation and traffic control to manage QoS and energy consumption in cloud computing environment. International Journal of Computer Applications in Technology, 60(4), 307-316.