JCBI
Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606
Volume 07 Issue 01
2024

Research Article
https://doi.org/10.56979/701/2024

Hybrid Approach at Cloud Data Center for Improving Makespan and Creating a Better Energy Enviornment

Noman Hasany[1], Waleed Younus[2], Fawad Salam Khan[3*], Abdul Hameed[4], and Raja M. Waqas Ahmed[4]

[1]Department of Software Engineering, Karachi Institute of Economics and Technology, Karachi, 75190, Pakistan.
[2]Director (IT & Procurement), Ministry of Finance, Government of Pakistan.
[3]Department of Creative Technologies, Faculty of Computing and AI, Air University, 44000, Islamabad, Pakistan.
[4]Department of Computer Science, Faculty of Computing and AI, Air University, 44000, Islamabad, Pakistan.
[*]Corresponding Author: Fawad Salam Khan. Email: fawad.salam@au.edu.pk

Received: February 22, 2024 **Accepted:** May 11, 2024 **Published:** June 01, 2024

Abstract: The field of cloud computing is growing quickly, and in order to achieve maximum performance and cost savings, effective resource management is required. The goal of this research is to adopt a hybrid technique to improve makespan in cloud data centers. In order to meet the increasing demand for cloud services, the main objective is to establish cost-effective and efficient cloud resource management. This work attempts to create a hybrid method, called HGWCA, by merging two different algorithms. The algorithms for Grey Wolf and Cat Swarm optimizations. The makespan, throughput, degree of imbalance, and turnaround time are the evaluation criteria that are employed. When compared to alternative algorithms, the suggested HGWCA method performs better in each of these metrics. The outcomes demonstrate that the hybrid strategy greatly enhances cloud data center performance based on makespan, degree of imbalance, throughput, and turnaround time. According to the study's findings, there is a lot of room for improvement in cloud data center performance with the suggested hybrid approach. Subsequent investigations could examine the suggested methodology in more extensive and intricate cloud data center setups, in addition to investigating the incorporation of extra optimization methods to enhance overall efficiency. The makespan improvement attained by the hybrid approach that was suggested was 5.18%. improvement in the result.

Keywords: Virtual machine, makespan, throughput, degree of imbalance, turnaround time.

1. Introduction

Many benefits come with cloud computing, including more collaboration, scalability, dependability, and affordable solutions. Nevertheless, there are potential drawbacks as well. First of all, accessing cloud services necessitates a steady internet connection, which can be difficult in places with spotty or inconsistent service. Second, due to data transmission issues and server congestion, cloud-based web applications may operate more slowly. Thirdly, because all of its components run online, cloud computing is susceptible to security lapses and intrusions. Finally, there's a chance that private information will be stolen or used illegally.

It's critical to take into account these disadvantages and evaluate if cloud computing is appropriate given the demands for connectivity, performance, and security [1-2]. In cloud computing, load balancing

is a technique for effectively allocating cloud environment virtual machine resources. It seeks to provide a fair distribution of workloads across the available servers or virtual machines in order to maximize the usage of these resources. Load balancing enables improved performance and resource efficiency in the cloud by dynamically distributing and reallocating computing resources based on demand, hence preventing the overflowing of individual servers. The essential technology for ensuring equitable task distribution and effective resource utilization in a cloud context is load balancing [3-4]. Load balancing is a crucial component needed to divide dynamic workloads among nodes in a cloud system. In cloud computing, efficient workload balancing results in higher user satisfaction and more efficient use of resources. Application in cloud computing According to [5-6] load balancing reduces latency in data transmission and reception. Static load balancing and dynamic load balancing are its two varieties. Static load balancing transfers incoming traffic and requests to the server that has the least amount of load relative to the other servers after first assessing the load on each server. Static load balancing seeks to minimize communication latency and increase reaction time [7-8].

Dynamic load balancing doesn't require any prior information; it only depends on the system as it is right now. Workload balancing in the cloud environment does not require any prior knowledge of the system. During execution, this approach allows for dynamic load distribution among multiple servers and distributes network traffic among them at runtime.

It may change the load of a server that is overloaded to other servers as it is being executed, which makes it a more effective method than static load balancing [9-10].

Software known as a virtual machine allows several operating systems to operate on a single computer system. It is made up of two operating systems: the host OS, which is installed on the actual computer, and the guest OS, which is installed within the virtual machine [11].

Virtual machines make it possible to install many operating systems at once by sharing system resources. But the fact that these resources are shared may have an effect on system speed. System virtual machines and process virtual machines are the two categories of virtual machines. Multiple operating systems are supported by system virtual machines, each of which runs a separate copy, while The idle procedure certain programs are run in a regulated environment using virtual machines. In general, running several operating systems or applications on a single computer system is made flexible and efficient by virtual machines [12-13].

2. Related Work

The competitive swarm optimizer (CSO) method has a major drawback: premature convergence, or the tendency to enter local optima. Finding a balance between the exploration and exploitation phases could help alleviate this issue. Consequently, the dynamic competitive swarm optimizer (DCSO) approach was developed, which modifies the seeking mode and selection scheme of the prior CSO algorithm. These modifications added dynamism to the algorithm and—above all—established a healthy equilibrium between the stages of exploration and exploitation. Additionally, the proportion of these phases was determined and this balance was further supported by the dimension-wise variety assessment. The findings show that, in the CSO algorithm, the ratio between the two phases is roughly 75% to 25%, whereas in the DCSO algorithm, there is a virtually equal fifty percent on average between these two stages. The experiment evaluated the robustness of the suggested approach against 33 benchmark functions and a real-world application-related backboard wiring issue [14].After that, the results are compared with three other popular methods. Consequently, the proposed approach yields highly competitive results and resolves the premature convergence issue. Still, there's always room for development. For example, the algorithm's

efficiency could be greatly increased by combining its seeking mode with a suitable local search strategy, as Golden Section search [15] [32].

As the power system has developed, the emphasis has switched to low-carbon, green operation, whereas the current power grid planning is mostly based on operating economics and reliability. This research suggests a cooperative grid planning method that promotes reliable and ecologically beneficial power grid operation, based on a modified CSO algorithm. A planning model is created that takes reliability, cost, and environmental aspects into account while examining the characteristics of carbon emissions during the building of a power system. Quantum theory and chaotic algorithms are used to improve the CSO algorithm, which efficiently solves the low-carbon planning model and supports a stable and sustainable power system [31]. The model experiment, which makes use of IEEE 39 bus technology, reveals that the suggested collaborative planning method has a construction cost of 23 million yuan and a carbon emission of 2.28 t/MWh. This approach lowers carbon emissions, maximizes building costs, and guarantees a stable, low-carbon power system operation [16-18].

3. Methodology

Initially, tasks are assumed to be $= \{T_1 + T_2 + T_3 \dots \dots .\}$. These jobs must be submitted to the task manager via the cloud console. In this setup, the task management module needs to ascertain the relative importance of each job, and then ascertain the relative importance of each virtual machine in terms of the power it consumes per unit of time. Figure 1 present the proposed model.

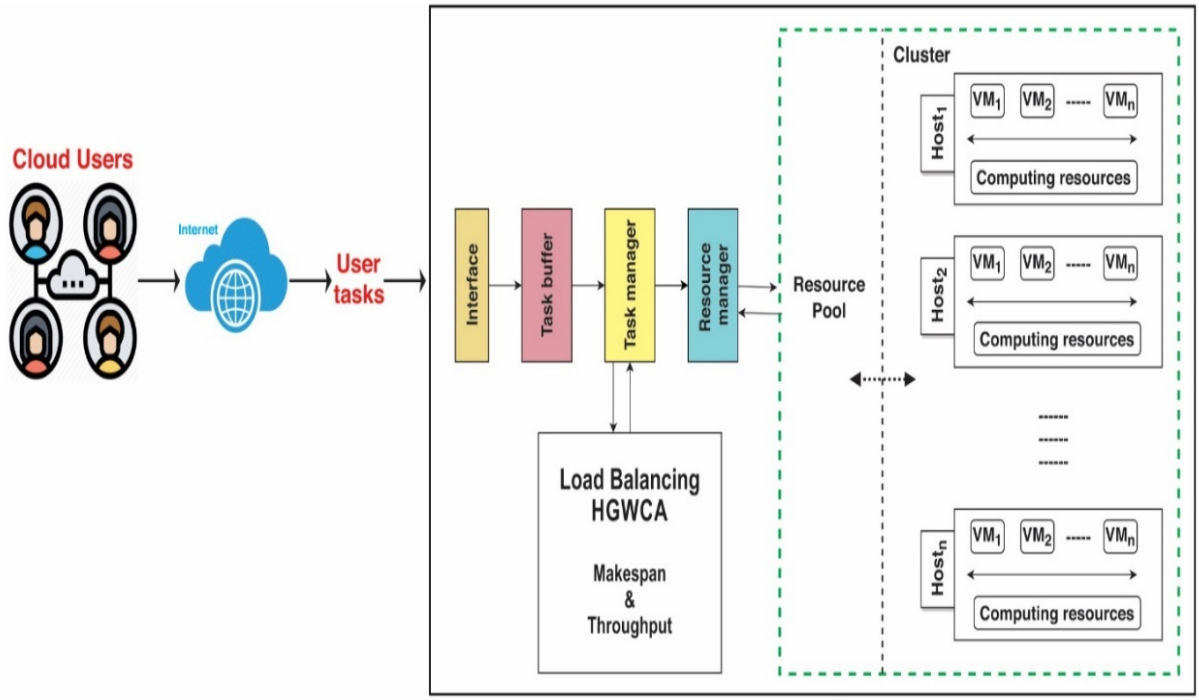

Figure 1. Proposed approach model

This is being tested because the jobs being brought into the cloud platform have many variations, and it's crucial to map them to the appropriate virtual resources in the cloud. The task scheduler is in charge of carrying out this task; a task scheduler connected to a resource manager module that keeps track of resource requests, resource allocations, and resource availability on the corresponding physical hosts. Within the data centers. This method assumes that n virtual machines (VMs) exist, with names such: Virtual machines (VMs) $V_n = \{V_1, V_2, V_3, \dots \dots V_n\}$ must live in physical hosts with name $H_i = \{H_1, H_2, H_3, \dots \dots H_i\}$ and the datacenters, also known as centers, where these hosts are kept, have names like$d_j = \{d_1, d_2, d_3, \dots \dots d_j\}$.The architecture as stated assigns tasks to virtual machines (VMs) based

on the relative cost of power in the datacenters, as determined by the Task Priority and VM Priority, respectively. A priority calculation must be performed in order to assign each task to the most appropriate virtual machine (VM), as each work has different processing requirements. The cloud's priority calculations for virtual machines (VMs) must take into consideration geographical variations in in electricity expenses [19-21]. A method that first identifies which tasks are most important, and then assigns those tasks to virtual machines that are best suited to executing those tasks using the least amount of electricity per unit of work, is used to reduce the overall power cost and energy consumption in cloud data centers. The purpose of the scheduler is to allocate jobs to virtual machines in a way that optimizes power consumption and expenses. The problem can be stated in terms of the tasks, virtual machines, physical hosts, and datacenters that were previously mentioned. It made a number of assumptions regarding tasks T_K, virtual machines V_n, physical hosts H_i, and datacenters d_j in order to schedule workloads onto virtual machines. In this configuration, it has to ascertain the the relative significance of virtual machines and workloads. These settings are examined and given a single priority when jobs are submitted to the task manager [22] [30]. The scheduler assigns jobs to the virtual machines (VMs), and in order to determine priorities, it is necessary to ascertain the VMs' present load. The computation of the load on every virtual machine is shown in the equation below.

$$L_h = \frac{L_v}{\sum_{k=1}^{m} H_k}$$

All virtual machines' (VMs') current load is represented by the variable L_h the capacity of the physical hosts must be ascertained after the load on each virtual machine (VM) has been computed, since every VM is meant to be housed within these hosts.

$$Total_{Powercost} = grid_{cost} + \left(dg_{cost} * a_{dg} \right) + green_{cost}$$

Virtual machines comprise L_h, the load on the physical host. In the cloud, a load balancing module is necessary. computing paradigm since VMs must either go to the next virtual machine on the same physical host by starting a new request or to the VM that is currently running if there are more tasks than they can handle. With a load balancer, this is feasible, but only after a specific threshold value is set to identify if the system is balanced or not.

This is how the system's threshold value is determine [23] [29].

$$energy^{con} = \sum energy^{con}(V_k)$$

Prior to determining whether the system's load is balanced, overloaded, or under loaded, the threshold value is established. The load balance is computed to arrive at this conclusion.

When a system is overloaded, it is said to be

$$energy^{con}(V_k) = \int_0^k energy_{comp}^{con}((V_k\ t) + energy_{idle}^{con}\ ((V_k\ , t)d_t$$

If the system is underloaded, it is said to be the system be Prior to determining whether the system's load is balanced, overloaded, or under loaded, the threshold value is established. The load balance is computed to arrive [24]-[28].

4. Results and Discussions

This section presents an analysis of a number of experimental results that were attained by using the suggested method. Several tests and investigations have been carried out with a variety of datasets and parameters. A detailed description of the experimental apparatus used in this work is given, offering important insights into the approach. A series of designated points clarify the overall procedure used in the tests. A crucial component of the plan's effective implementation is the availability of appropriate

datasets, which were obtained for this investigation from the WS. DREAM database. A modified CloudSim platform was used to construct the HGWCA method, and test datasets with file sizes ranging from 200 to 400 KB were used. The datasets were compliant with the standard workload format (SWF). In order to assess the HGWCA algorithm's performance, several Virtual machines (VMs) and jobs (200–2000) spread across many data centers in the cloud-computing environment. The HGWCA algorithm's efficacy was evaluated in comparison to other well-known algorithms, including ABC, MBat, HHO-ACO, and QMPSO, with an emphasis on important metrics for load balancing effectiveness, including makespan, throughput, turnaround time, and degree of imbalance. The comparison study showed that the suggested HGWCA method outperformed other algorithms with an accuracy rate of 1.25% and a notable makespan reduction of 0.98%. These results validate the feasibility and efficacy of the HGWCA algorithm in handling load-balancing issues in cloud computing settings. Table 1 present the complete no. of task of different algorithm based on no. of task during the testing.

Table 1. Complete No. of Task of Different Algorithm Based on No. of Task

Given No of Task	PSO	ABC	PSO-CALBA	M-Bat	HHO-ACO	Proposed HGWCA
200	160	165	170	175	186	193
400	355	365	372	378	385	390
600	560	570	575	580	588	592
800	750	770	775	780	786	791

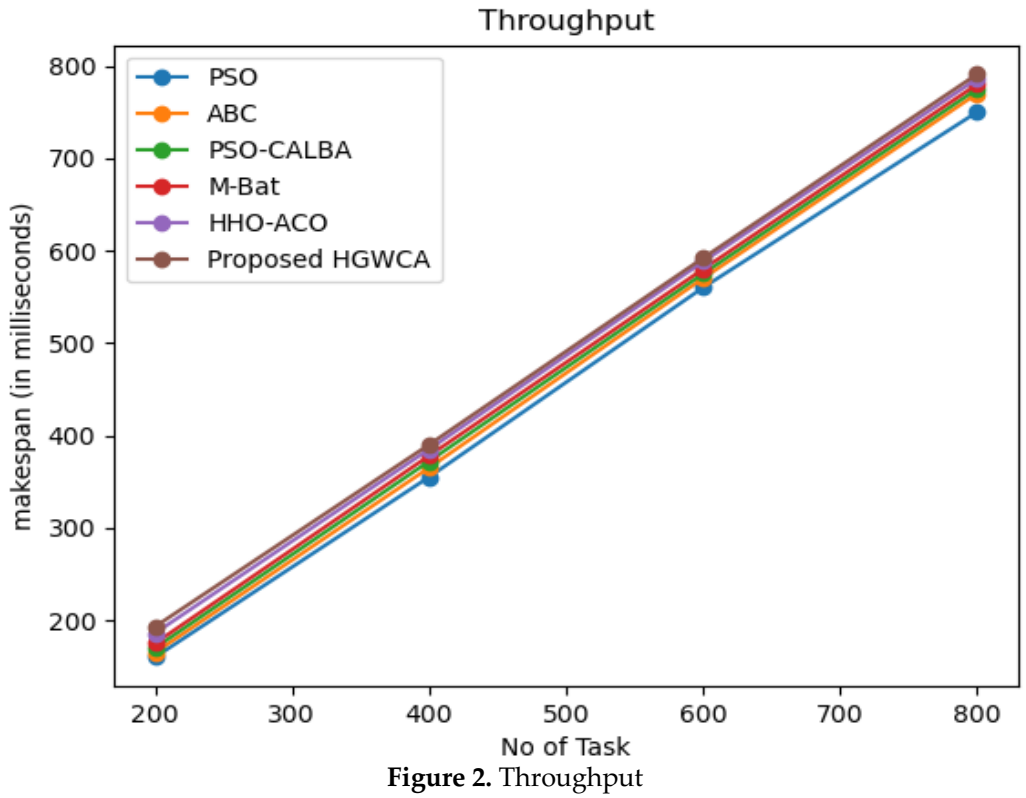

Figure 2. Throughput

Table 2. Makespan of Different Algorithm Based on no. of Virtual Machine

No. of VM	PSO	ABC	PSO-CALBA	M-Bat	HHO-ACO	Proposed HGWCA

10	48	48	47	46	46	45
20	100	98	97	97	97	96
30	198	196	195	194	194	192
40	398	395	393	390	390	388

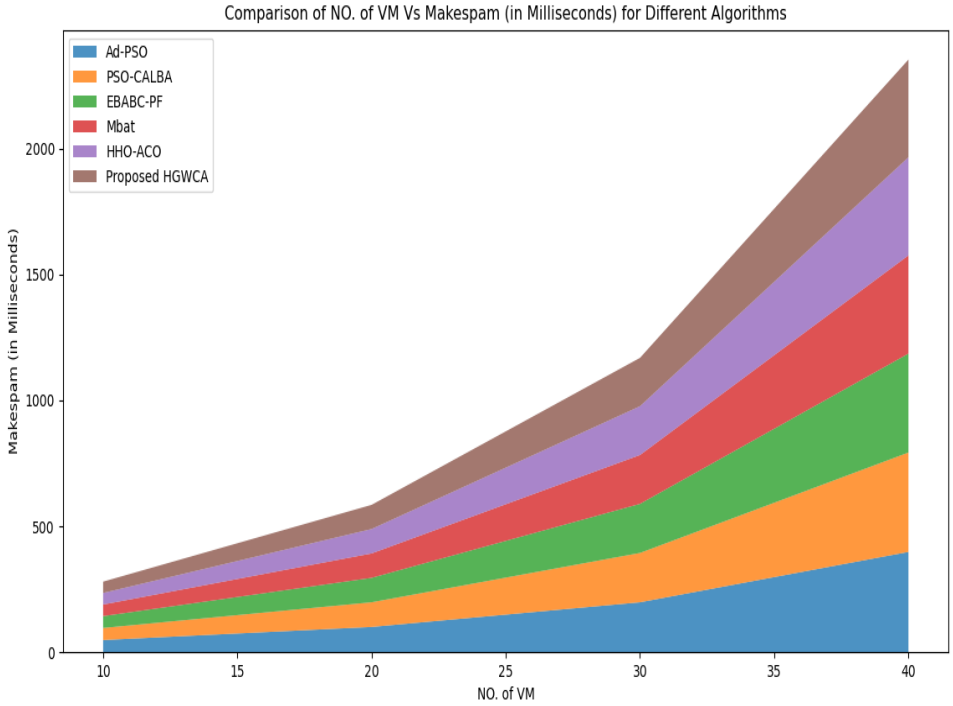

Figure 3. Comparison No. of VM Vs Makespan for Different Algorithms

Table 3. Makespan Based on No. of Task

Algorithm Name	Complete No of Tasks	Makespan (In Milli second)	Complete No of Tasks	Makespam (In Milli second)	Complete No of Tasks	Makespan (In Milli second)	No of Tasks	Makespan (In ms)
PSO	50	50	100	100	150	150	180	170
ABC	50	49	100	98	150	148	180	165
PSO-CALBA	50	47	100	94	150	147	180	162
M-Bat	50	47	100	92	150	146	180	157
HHO-ACO	50	45	100	92	150	144	180	155
Proposed HGWCA	50	43	100	90	150	142	180	152

The "Proposed HGWCA" technique is the best one based on the data in Table 1 to 3 and Figure 2 to 4 since it consistently achieves the lowest makespan values for varying numbers of virtual machines [27]. With consistently greater makespan values than other algorithms, the "PSO" algorithm is the worst. This suggests that when it comes to minimizing execution time, the "Proposed HGWCA" algorithm performs

the best, while the "PSO" approach performs the worst [28]. Figure 5 present the proposed algorithm result with standard algorithm taking different parameters. Based on the given result the proposed algorithm improve in different section of cloud computing like makespan, throughputs and degree of imbalance.

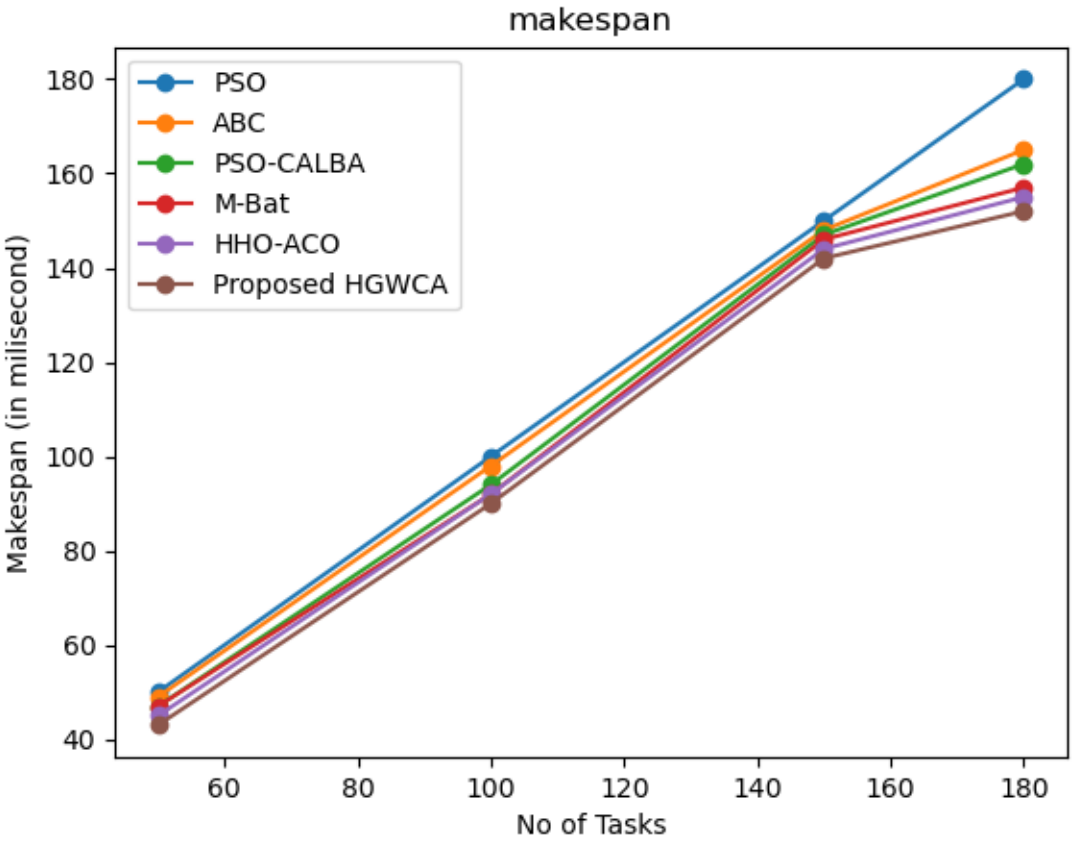

Figure 4. Makespan Based on No. of Task

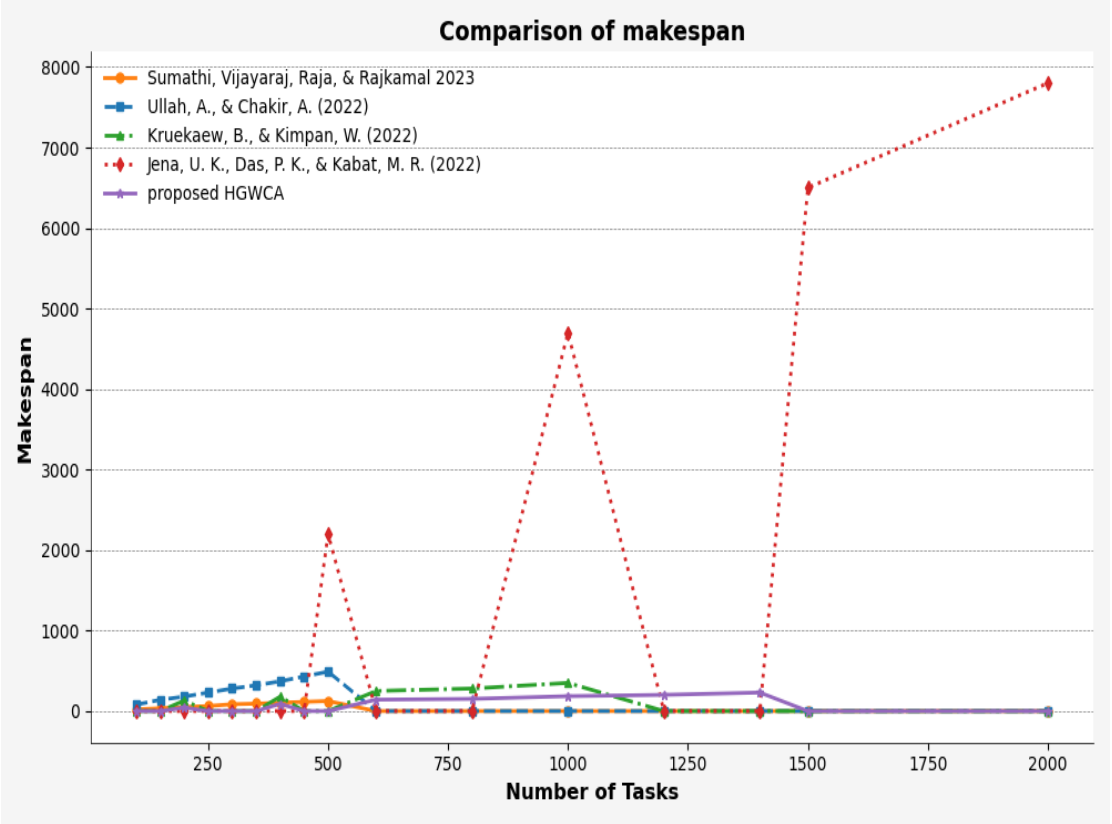

Figure 5. Comparison of Makespan with Existing Studies

5. Conclusion and Future work

There have been encouraging results from using the Grey Wolf Optimizer (GWO) and Cat Swarm Optimization (CSO) algorithms to increase makespan and throughput in cloud data centers. Reducing makespan and increasing throughput can be achieved by optimizing resource allocation and task scheduling through the integration of these sophisticated algorithms. The study's conclusions demonstrate how well GWO and CSO explore and utilize the solution space while taking into account a number of cloud-specific considerations. These algorithms can increase the productivity and customer satisfaction of cloud data centers by assigning resources and scheduling jobs in an intelligent manner. It is possible that more study in this field will advance optimization methods and support ongoing development of cloud datacenters [29] [30].

References

1. Bin, N. I. N. G., Qiong, G. U., Zhao, W. U., Lei, Y. U. A. N., & Chun-yang, H. U. (2015). Bats algorithm research in cloud computing resource scheduling based on membrane computing. Application Research of Computers/Jisuanji Yingyong Yanjiu, 32(3).
2. Ullah, A., Yasin, S., & Alam, T. (2023). Latency aware smart health care system using edge and fog computing. Multimedia Tools and Applications, 1-27.
3. Ullah, A., Khan, S. N., & Nawi, N. M. (2023). Review on sentiment analysis for text classification techniques from 2010 to 2021. Multimedia Tools and Applications, 82(6), 8137-8193.
4. Sebai, D., & Shah, A. U. (2023). Semantic-oriented learning-based image compression by Only-Train-Once quantized autoencoders. Signal, Image and Video Processing, 17(1), 285-293.
5. Ganne, A. (2022). Emerging Business Trends in Cloud Computing. International Research Journal of Modernization in Engineering Technology, 4(12).
6. Gundu, S. R., Panem, C. A., Thimmapuram, A., & Gad, R. S. (2022). Emerging computational challenges in cloud computing and RTEAH algorithm based solution. Journal of Ambient Intelligence and Humanized Computing, 1-15.
7. Alam, T., Gupta, R., Qamar, S., & Ullah, A. (2022). Recent applications of Artificial Intelligence for Sustainable Development in smart cities. In Recent Innovations in Artificial Intelligence and Smart Applications (pp. 135-154). Cham: Springer International Publishing.
8. Ullah, A., & Chakir, A. (2022). Improvement for tasks allocation system in VM for cloud datacenter using modified bat algorithm. Multimedia Tools and Applications, 81(20), 29443-29457.
9. Kumar, R., Bhardwaj, D., & Joshi, R. (2022). Adaptive bat optimization algorithm for efficient load balancing in cloud computing environment. In Advances in Computational Intelligence and Communication Technology: Proceedings of CICT 2021 (pp. 357-369). Singapore: Springer Singapore.
10. Li, X., Lu, Y., Fu, X., & Qi, Y. (2021). Building the Internet of Things platform for smart maternal healthcare services with wearable devices and cloud computing. Future Generation Computer Systems, 118, 282-296.
11. Nadimi-Shahraki, M. H., Taghian, S., & Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications, 166, 113917.
12. Nadimi-Shahraki, M. H., Taghian, S., & Mirjalili, S. (2021). An improved grey wolf optimizer for solving engineering problems. Expert Systems with Applications, 166, 113917.
13. Panwar, K., & Deep, K. (2021). Discrete Grey Wolf Optimizer for symmetric travelling salesman problem. Applied Soft Computing, 105, 107298.
14. Krishnamoorthy, P. (2021). Performance Analysis of Hybrid BAT Algorithm and Cuckoo Search Algorithm [HB-CSA] for Task Scheduling in Mobile Cloud Computing. Available at SSRN 3997784.
15. Gundu, S. R., Panem, C. A., & Thimmapuram, A. (2020). Hybrid IT and multi cloud an emerging trend and improved performance in cloud computing. SN Computer Science, 1(5), 256.
16. Ullah, A., Nawi, N. M., & Khan, M. H. (2020). BAT algorithm used for load balancing purpose in cloud computing: an overview. International Journal of High Performance Computing and Networking, 16(1), 43-54.
17. Ibrahim, L. M., & Saleh, I. A. (2020). A solution of loading balance in cloud computing using optimization of bat swarm algorithm. Journal of Engineering Science and Technology, 15(3), 2062-2076.
18. Chung, K., & Park, R. C. (2019). Chatbot-based heathcare service with a knowledge base for cloud computing. Cluster Computing, 22, 1925-1937.
19. Panda, M., & Das, B. (2019). Grey wolf optimizer and its applications: a survey. In Proceedings of the Third International Conference on Microelectronics, Computing and Communication Systems: MCCS 2018 (pp. 179-194). Springer Singapore.

20. Punitha, A. A. A., & Indumathi, G. (2019). Centralized cloud information accountability with bat key generation algorithm (CCIA-BKGA) framework in cloud computing environment. Cluster Computing, 22(Suppl 2), 3153-3164.
21. Jian, C., Chen, J., Ping, J., & Zhang, M. (2019). An improved chaotic bat swarm scheduling learning model on edge computing. IEEE Access, 7, 58602-58610.
22. Patil, R., Dudeja, H., & Modi, C. (2019). Designing an efficient security framework for detecting intrusions in virtual network of cloud computing. Computers & Security, 85, 402-422.
23. Chauhan, R., & Kumar, A. (2013, November). Cloud computing for improved healthcare: Techniques, potential and challenges. In 2013 E-health and bioengineering conference (EHB) (pp. 1-4). IEEE.
24. FS. Khan, T Naqash, MI Khatak, RM Larik (2015, Nov). LIGHT AND SECURE COMMUNICATION ALGORITHM FOR COGNITIVE RADIO NETWORK BY USING LABYRINTHINE AUTHENTICATION. In Jurnal Teknologi 76 (1)
25. FS Khan, N Hasany, A Altaf, MNA Khan, Arifullah Benchmarking of an Enhanced Grasshopper for Feature Map Optimization of 3D and Depth Map Hand Gestures . In Journal of Computing & Biomedical Informatics 7 (1), 1-8
26. FS Khan, MNH Mohd, SABM Zulkifli, GE Mustafa, SK Abro, DM Soomro Deep Reinforcement Learning Based Unmanned Aerial Vehicle (UAV) Control Using 3D Hand Gestures, In CMC-Computers, Material & Continua 72 (3), 5741-5759
27. FS Khan, MNH Mohd, DM Soomro, S Bagchi, MD Khan 3D hand gestures segmentation and optimized classification using deep learning. In IEEE Access 9, 131614-131624
28. Shaker, B., Ullah, K., Ullah, Z., Ahsan, M., Ibrar, M., & Javed, M. A. (2023, November). Enhancing grid resilience: Leveraging power from flexible load in modern power systems. In 2023 18th International Conference on Emerging Technologies (ICET) (pp. 246-251). IEEE.
29. Munir, A., Sumra, I. A., Naveed, R., & Javed, M. A. (2024). Techniques for Authentication and Defense Strategies to Mitigate IoT Security Risks. Journal of Computing & Biomedical Informatics, 7(01).
30. Ali, H., Iqbal, M., Javed, M. A., Naqvi, S. F. M., Aziz, M. M., & Ahmad, M. (2023, October). Poker Face Defense: Countering Passive Circuit Fingerprinting Adversaries in Tor Hidden Services. In 2023 International Conference on IT and Industrial Technologies (ICIT) (pp. 1-7). IEEE.
31. Khan, M. F., Iftikhar, A., Anwar, H., & Ramay, S. A. (2024). Brain Tumor Segmentation and Classification using Optimized Deep Learning. Journal of Computing & Biomedical Informatics, 7(01), 632-640.
32. Ali, A. S., Iqbal, M. M., Khan, A. H., Hameed, N., & Bibi, S. (2023). Lung Cancer Detection Using Convolutional Neural Networks from Computed Tomography Images. Journal of Computing & Biomedical Informatics, 6(01), 133-143.