iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** VII **Month of publication:** July 2023

DOI: https://doi.org/10.22214/ijraset.2023.54710

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue VII Jul 2023- Available at www.ijraset.com

Improved Task Scheduling for Cloud Computing Using Combination of Genetic Algorithm and Tabu Search Algorithm

Supriya Singh[1], Jitendra Agrawal[2]
[1]*SoIT UTD RGPV, Bhopal Madhya Pradesh, India*

Abstract: ***Cloud computing has revolutionized the way computational resources are provisioned and utilized. However, effective task scheduling in cloud environments remains a critical challenge due to the complexity of managing resources and meeting quality of service (QoS) requirements. In this paper, we address the multi-objective task scheduling problem by proposing a novel scheme that combines a hybrid algorithm, HGTSA, to achieve improved convergence rates and overall efficiency. Our main contributions include the development of a fitness function that considers makespan, system capacity, and resource utilization, as well as the implementation of the HGTSA algorithm. Through extensive simulations, we demonstrate the superiority of our proposed scheme, outperforming conventional genetic algorithms and achieving better makespan times. Furthermore, we highlight the future scope of research in leveraging machine learning techniques to further enhance task scheduling processes and optimize decision-making based on workload predictions and historical data.***
Keywords: Task Scheduling, Optimization Algorithms, Cloud Computing, Distribution systems, Cloud Applications etc.

I. INTRODUCTION

Cloud computing has emerged as a transformative technology, catering to the growing need for scalable and flexible computing resources in various industries. The ability to access and utilize vast computational power and storage capabilities offered by cloud providers has revolutionized the way organizations handle their data and applications [1]. However, with the increasing scale and complexity of cloud systems, efficient management and allocation of tasks have become critical for achieving optimal performance and resource utilization. Distribution systems, within the context of cloud computing, involve the allocation and execution of tasks across multiple nodes or machines in the cloud infrastructure. The distribution of tasks allows for parallel processing and enables faster execution of computationally intensive workloads. However, the effectiveness of task distribution heavily relies on intelligent and dynamic task scheduling algorithms.

Task scheduling plays a pivotal role in distribution systems, where the goal is to assign tasks to appropriate resources in an efficient and optimized manner [2]. It involves making decisions regarding task placement, considering factors such as resource availability, workload balancing, energy efficiency, and meeting user-defined quality of service (QoS) requirements. Effective task scheduling ensures that tasks are executed in a timely manner, resource utilization is optimized, and system performance is maximized. Efficient task scheduling in distribution systems brings numerous benefits to cloud computing environments. Firstly, it enables improved resource utilization by effectively mapping tasks to available resources, minimizing resource wastage, and ensuring optimal use of computational power [3]. This, in turn, leads to cost savings for cloud service providers and enhanced scalability for users. Secondly, effective task scheduling helps in achieving better system performance and response times. By distributing tasks intelligently, the workload is balanced across the cloud infrastructure, avoiding resource bottlenecks and reducing the chances of system overload. This results in faster task execution and improved user experience.

A. Factors and Recent Technologies

Several factors significantly impact the performance of task scheduling in cloud computing environments. These factors arise from the dynamic nature of cloud systems, varying workload demands, resource heterogeneity, and the diverse quality of service (QoS) requirements of users. Traditional static scheduling algorithms often struggle to cope with these complexities and fail to deliver optimal solutions [4]. To overcome these challenges, recent advancements in task scheduling have introduced innovative technologies that leverage advanced optimization algorithms. These algorithms employ mathematical and heuristic techniques to explore and identify the most optimal scheduling solutions, considering multiple objectives and constraints [5].

By accounting for factors such as task dependencies, resource availability, workload distribution, and energy consumption, these optimization algorithms aim to achieve efficient and effective task scheduling. Prominent among these optimization algorithms are genetic algorithms, particle swarm optimization, ant colony optimization, and simulated annealing. These algorithms possess the capability to adapt to dynamic environments, handle intricate scheduling scenarios, and provide near-optimal solutions [6]. By employing these technologies, the efficiency and effectiveness of task scheduling in distribution systems within cloud computing environments can be significantly improved. The utilization of optimization algorithms enables intelligent decision-making, facilitating workload balancing, resource utilization optimization, and meeting QoS requirements [7]. The application of optimization algorithms in task scheduling presents an opportunity to enhance the performance of cloud systems. These algorithms can dynamically adjust task assignments, adapt to changing conditions, and optimize resource allocation, ultimately leading to improved system efficiency and user satisfaction [8]. By harnessing the power of these recent technologies, cloud computing providers can better meet the ever-growing demands of their users and enhance the overall performance of distributed task scheduling in cloud environments.

Inspired from this, this article presents a pioneering approach to task scheduling in cloud computing by introducing a novel optimization algorithm capable of addressing NP-hard problems, including the intricate task scheduling challenges. Notably, the proposed scheme sets itself apart by considering multiple quality factors, a significant departure from traditional approaches commonly found in recent research. The subsequent sections of this article delve into a comprehensive literature review of the existing task scheduling solutions, highlighting the growing prominence of optimization algorithms and shedding light on the current gaps in the field. Furthermore, the methodology and materials utilized in this research are meticulously defined, providing insights into the experimental framework. The obtained results from the simulations are thoroughly discussed and analyzed. In conclusion, this study offers valuable contributions to the domain of task scheduling in cloud computing, presenting a robust optimization algorithm and underscoring the importance of considering multiple quality factors. The findings not only contribute to the existing body of knowledge but also provide a solid foundation for future research endeavors in this area.

II. LITERATURE SURVEY

In recent years, task scheduling in cloud computing has garnered significant attention from researchers and practitioners alike. This section presents an overview of recent advancements in task scheduling within the context of cloud computing. The focus is on exploring the work conducted by various authors in this domain, specifically with regards to the utilization of optimization algorithms. By examining the current methods employed by researchers, this study aims to provide a comprehensive understanding of the subject matter.

G. Saravanan et al. [9] highlight the introduction of the an Improved variant of Wild Horse Optimization algorithm to be an option for tackling the issues of complicated scheduling times, expensive, and high load at different VMs in cloud computing task scheduling within a particular of their research proposals. The proposed algorithm, IWHO, offers an improved approach to enhance scheduling efficiency in cloud environments.

Sudheer Mangalampalli [10] proposed an efficient task scheduling algorithm that prioritizes both tasks and virtual machines, ensuring accurate allocation of tasks to appropriate VMs. The approach, which uses firefly optimization, is assessed using real-time worklogs from HPC2N and NASA as well as constructed information sets with various variances. By enhancing accuracy as well as efficiency of systems via priority-based scheduling, this method advances the area of cloud-based task allocation.

Peng, Zhihao, et al. [11], suggested a parallel GA using a MapReduce structure to schedule tasks on cloud computing with a variety of priority queues. This study's main objective was to use MapReduce architecture to reduce the amount of time it takes for the task allocation process to complete overall in a cloud-based computing environment. this research presented a groundbreaking approach to task scheduling by combining genetic algorithms with heuristic methods and integrating them with the MapReduce framework. The proposed technique surpassed previous algorithms and demonstrated superior performance in allocating tasks to processing units and assigning tasks to processors in heterogeneous resource environments, ultimately enhancing overall system efficiency.

Sanaj, M. S., et al. [12], proposed an effective method for a scheduler in a particular cloud using the MAP reduction framework and GA-WOA. The client's task was first used to retrieve the task characteristics. The MRQFLDA method was then used to decrease the characteristics. The massive tasks were then split up into more manageable ones using a map-reduce framework. Finally, the task was successfully organized using the GA-WOA method. To execute the simulated simulations, the cloud infrastructure was utilized. The findings demonstrated that the suggested approach GA-WOA outperforms the other methods in terms of the different metrics assessed.

Ramezani, F et al. [13], provides a comprehensive multi-objective modelling framework for scheduling methods that balances several competing objectives including time, power usage, etc. From both the provider's and the client's perspectives, this technique reduces costs by accounting for performance and electrical costs. The suggested framework is assessed using the GA and PSO with multi-objective. The simulation findings demonstrate that the suggested multi-objective model identifies the best trade-offs between the four competing objectives, greatly reducing the reaction time and makespan of the work. This model lowers the cost to providers while simultaneously improving QoS. According to the findings of our experiments, MOPSO was a more rapid and precise evolutionary algorithm than MOGA for resolving similar issues.

Weiqing, G. E., et al. [14], introduced min-min and max min based algorithm task scheduling techniques in which task completion time and load balancing factors were selected for double fitness values. Though the simulation results of the new algorithm still need to be better, the study completes the initial fusion of the technique and raises the possibility of merging the two qualitatively improved algorithms.

Nie, Jun. et al. [15], analyzed various task scheduling techniques and programming models used in CC and proposed a Genetic algorithm and ACO-based hybrid algorithm. This technique overcomes the early pheromone lacking in the ant colony, which leads to a slow resolution, and fully utilizes the rapid random global search capability of genetic algorithm. The simulation results illustrate that this approach executes well and increases work scheduling effectiveness in a cloud computing environment. It was effective in a cloud computing environment.

Wu, Z et al. [16], suggested an approach based on the GA and ACO. This task scheduling technique can facilitate faster user job completion on the cloud system. The simulator experiment findings in CloudSim demonstrate that the proposed approach outperforms the genetic algorithm and ant colony optimization algorithm individually in terms of load management and ideal period.

Khan, M.S.A., et al. [17], presented a hybrid optimization-based task scheduling system that efficiently organizes tasks with the least amount of waiting time. The simulation findings demonstrate that the suggested scheduling method outperforms traditional scheduling algorithms based on ant colony and swarm optimization.

Li, Yuxia. Et al. [18], incorporates the symbiotic algorithm with the classic ACO algorithm to enhance its low workflow scheduling effectiveness in cloud computing. The ant colony algorithm was first divided into two subgroups, and the symbiotic algorithm's symbiosis, cohabitation, and parasite processes are utilized to keep the algorithm from entering a local optimum and speed it up to reach accurate solutions. Because of this, the outcomes of the cloud computing workflow scheduling simulator demonstrate that the proposed algorithm successfully increases cloud computing task scheduling efficiency.

Guo, Qiang. et al. [19], suggested a cloud computing task scheduling method based on the ant colonies algorithm to maximize the task scheduling strategy in a cloud environment. This algorithm's primary objective was to reduce task duration and cost overall while balancing the workload on the system. In this essay, we describe the load balance function and build the objective functions for job duration and costs. In terms of makespan, expenses, and system load balancing, the results demonstrate that the method is more effective than the other two algorithms.

Despite significant advancements in task scheduling for cloud computing, there remain several notable research gaps that need to be addressed. Firstly, while optimization algorithms have gained popularity in recent years, there is still a need for more tailored algorithms specifically designed to address the complexities of task scheduling in cloud environments. The existing algorithms often struggle to handle dynamic workload variations, resource heterogeneity, and diverse quality of service requirements effectively. Secondly, the consideration of multiple quality factors in task scheduling is an area that requires further exploration. Although some recent approaches have started incorporating multiple factors, there is a lack of comprehensive frameworks that integrate various quality aspects, such as capacity, resource utilization, completion time, etc. into a unified scheduling model. Addressing these research gaps will contribute to the development of more robust and efficient task scheduling approaches, enabling better resource utilization, improved system performance, and enhanced user satisfaction in cloud computing.

III. PRESENT WORK

In the current research, an optimization algorithm has been implemented by combining the methodologies of Genetic Algorithm (GA) and Tabu Search Algorithm. This fusion of approaches aims to overcome the limitations of each algorithm and achieve high-performance task scheduling in cloud computing. The primary motivation behind this combination is to leverage the strengths of both algorithms. GA, with its ability to explore a broad search space, provides diversity in solutions, while Tabu Search Algorithm, with its higher convergence rate, accelerates the optimization process. By synergistically utilizing these algorithms, the proposed model strives to improve the overall efficiency and effectiveness of task scheduling.

Furthermore, the proposed model enhances the fitness function of the optimization algorithms by incorporating considerations of resource utilization and VM capacity. Traditionally, fitness functions focus on single objectives, such as makespan or energy consumption. However, in this research, the model takes a more comprehensive approach by considering multiple factors that impact task scheduling performance. By incorporating resource utilization and VM capacity and makespan time into the fitness function, the proposed model aims to optimize the allocation of tasks to resources, ensuring efficient utilization of available resources and accommodating the capacity constraints of VMs. This integrated approach contributes to the overall enhancement of task scheduling performance in cloud computing environments. The detailed description with algorithm and figures are discussed further in this section.

A. Methods and Simulation Setup

In this section, the proposed approach for task scheduling in a distributed system with heterogeneous processors and a high-speed network is outlined. The methodology involves the random generation of diverse task configurations, which are subsequently evaluated using the proposed mechanism. It is assumed that each subtask can be executed on a single processor, and the communication cost between linked subtasks is considered zero when scheduled on the same processor. Notably, the utilization model adopted in this method is unrelated, allowing for variations in processing time across different activities performed by a processor.

To validate the efficacy of the proposed approach, a simulation is conducted using three distinct task setups: one comprising 20 tasks, another with 40 tasks, and a third with 80 tasks. These task setups are selected to encompass a range of complexities and system loads, enabling a comprehensive evaluation of the performance of the proposed task scheduling mechanism. The simulation aims to analyze and compare the efficiency, resource utilization, and overall system performance achieved by the proposed approach in the context of the distributed system. By leveraging these specific methods and materials, this study aims to provide valuable insights into optimizing task scheduling in a distributed system with parallel processors. The simulations for this study were conducted using the MATLAB software on a Windows 10 64-bit operating system. The computer system used had a configuration of 8 GB RAM and 500 GB of storage space. In order to implement the objectives of the proposed model and optimization algorithm, a thorough understanding of the underlying concepts was essential to develop an effective optimization approach. Further details and information regarding the proposed algorithm will be discussed in subsequent sections, where the algorithm's design and functioning will be elaborated upon

IV. PROPOSED SCHEDULING ALGORITHM

The proposed framework for task scheduling is depicted in Figure 1, providing a visual representation of the system architecture and the flow of the scheduling process. This framework draws inspiration from the Genetic Algorithm (GA) and incorporates the search concepts of Tabu Search. By combining the strengths of GA, which excels in exploring a wide search space, with the faster convergence rate of the Tabu Search Algorithm, the proposed model aims to overcome the limitations of each individual algorithm named as Hybrid Genetic Taboo Search Algorithm (HGTSA), resulting in improved overall optimization efficiency in comparison to other existing approaches.

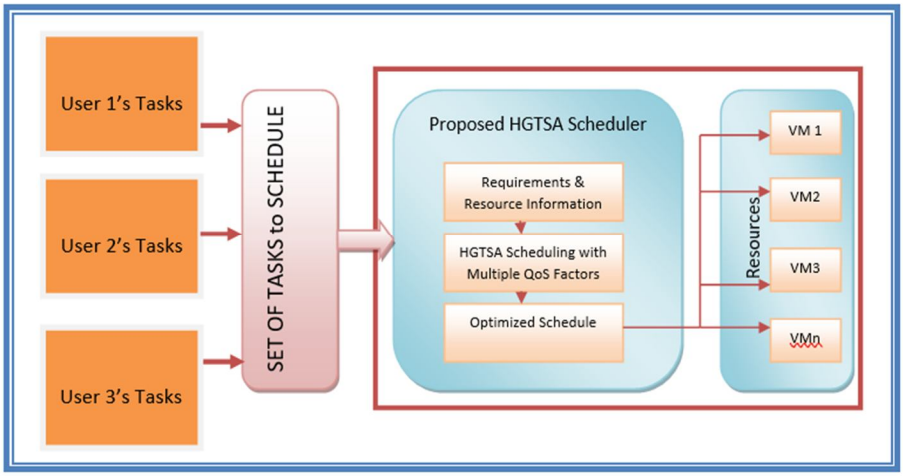


Fig 1. Proposed Framework for scheduling

1) *Initial Population Generation:* In this application, the initial step is to define the initial information for the setup, which includes the number of tasks, number of CPUs, and other relevant parameters. The details of all these parameters can be found in Table 1 below. Once the configuration of the cloud environment is defined, the next step is to initialize the population for the application. For the simulation, three different configurations are considered, representing distinct scenarios. To introduce dynamism and make the code more adaptable, the processing times of the VMs are randomly generated. This randomization allows for a more realistic and dynamic representation of the system. The initial population of the system is then generated based on the number of tasks to be scheduled using the proposed algorithm.

Table 1: System setup configuration details

Parameter	Values
No of CPU	2
MIPS	250
Bandwidth	1000
No. of Tasks	20,40,80
Population	10
Iteration	120
Crossover Rate	Random [0.5 0.8]
Mutation Rate	Random [0.5 0.8]

By following this approach, the application ensures that the initial population represents different task scheduling possibilities within the given cloud environment, thereby setting the stage for the subsequent execution of the proposed algorithm and the optimization process.

2) *Fitness Function:* In the proposed scheme, the fitness function plays a crucial role in determining the suitability of chromosomes for the next generation in the task scheduling process. This scheme introduces a novel contribution to the task scheduling domain by incorporating multiple Quality of Service (QoS) factors that are rarely considered collaboratively. The fitness function, as depicted in Equation 1, combines three essential factors: makespan time, system capacity, and resource utilization.

$$fitness = f_{makespan} + f_{capacity} + f_{Ru} \text{---------- (1)}$$

Where, $f_{makespan}$, $f_{capacity}$ and f_{Ru} are the calculated values for makespan time, system's capacity and resource utilization given by as follow.

$$f_{capacity} = \frac{\text{MIPS} \times \text{CPU}_{\text{count}}}{BW} \text{---------- (2)}$$

$$f_{makespan} = \max\{ET_i\} \text{---------- (3)}$$

$$f_{Ru} = \frac{ET_i}{f_{makespan}} \text{---------- (4)}$$

Where, BW is bandwidth and ET_i is task execution time.
By combining these QoS factors in the fitness function and providing explicit formulations for each factor, the proposed scheme aims to optimize task scheduling by considering multiple performance metrics. This approach enables a more comprehensive evaluation of chromosomes, resulting in improved task allocation decisions based on a holistic assessment of the system's performance.

3) *Crossover and Mutation:* The crossover and mutation operators play crucial roles in the proposed scheme for task scheduling. Crossover involves exchanging genetic material between parent chromosomes to generate offspring, promoting exploration of the search space. Mutation introduces random changes in individual chromosomes, adding diversity to the population. These genetic operators enable the algorithm to explore new solutions, improve the quality of schedules, and enhance overall performance. In proposed scheme crossover and mutation rate is varied in every iteration ranging between 0.5 to 0.8 in order to avoid static solution and repeating chromosomes.

4) *Selection of Solution:* The suggested approach employs a time-efficient roulette wheel selection technique with random acceptance to choose chromosomes for the local search having complexity O(1). It has 2 crucial steps to implement it. Firstly, a chromosome is randomly selected from the population using a selection probability of 1/n. In the second step, the fitness value of the selected chromosome is analyzed to determine if it is one of the elites. If it is not, Tabu search is employed to find a new solution. The overall algorithm for finding the optimum solution is presented in Algorithm 1.

5) *Termination of Algorithm:* The termination condition for the given scheme is based on the completion of a specified number of iterations. The algorithm continues executing until the desired number of iterations is reached. At each iteration, the genetic operators, such as crossover, mutation and tabu search, are applied to the population of chromosomes, and the fitness of the resulting offspring is evaluated. This iterative process allows the algorithm to evolve and improve the quality of solutions over time. Once the predetermined number of iterations is completed, the algorithm terminates, and the best solution obtained thus far is considered the final result.

Algorithm 1

- Input:
  - Size of Population to be generated, n
  - Total Number of iterations to process, Iter
- Algorithm's outcomes:
  - best schedule for task scheduling
- Generate a population as an initial schedule for n tasks.
- Initialize the tabu list TL
- Set the counter t = 0 representing iterations count.
- Evaluate the initial fitness of population generated initially using equation (1)
- Get f_{best} as initial best
- While (t <Iter):
  - Selection of a solution set as per their fitness values.
  - Apply step for crossover over selected solution.
  - Calculate fitness after crossover
  - if $fitness<f_{best}$
    - get better then update f_{best}
  - Mute the solution using random generated mutation probability.
  - Calculate fitness after crossover
  - if $fitness<f_{best}$
    - get better then update f_{best}
  - else
    - Update schedules with tabu list
  - Replace the old solution set with new one.
  - Counter t is to increment.
- Select the best solution (schedule) from the final population.
- End.

Moving forward, the following section will delve into the discussion and analysis of the obtained results.

V. SIMULATION RESULTS

This section aims to provide insights into the performance and effectiveness of the proposed scheme for task scheduling in cloud computing. The results will be examined and interpreted, taking into account various performance metrics, such as makespan time, system capacity, and resource utilization. The analysis will highlight the strengths and limitations of the proposed scheme and shed light on its potential implications and future directions for further improvement.

The performance factor considered for evaluation is makespan time. The simulation is conducted in MATLAB and is analyzed with different existing algorithms including IWD, PSO, GA, it also included WOA and MFO. The outcomes are analyzed with best, medium and worst solution given by proposed method.

As shown in Fig2 depicts the comparison of proposed HGTSA algorithm under best condition where the task was 20. The proposed HGTSA algorithm achieved the lowest makespan time of 231, while IWD and PSO took 280 and GA took 251. The suggested algorithm performed better due to the formed a task scheduling mechanism that was more effective and efficient and took resource consumption and VM capacity into account in the fitness measure. The outcomes show that the suggested HGTSA algorithm offers a potentially viable option for practical and effective scheduling of tasks in cloud computing environments.

Fig 3 compares the makespan time of different optimization algorithms for task scheduling under best case where task was 40. The proposed HGTSA algorithm achieved the lowest makespan time of 242, outperforming other algorithms such as PSO [11], GA [11], WOA [11], etc. The efficient GA algorithm, along with the incorporation of resource consumption and VM capacity considerations in the fitness function, led to the superior performance of the proposed algorithm. These results demonstrate the potential of the HGTSA algorithm for efficient and effective task scheduling in cloud computing under best condition

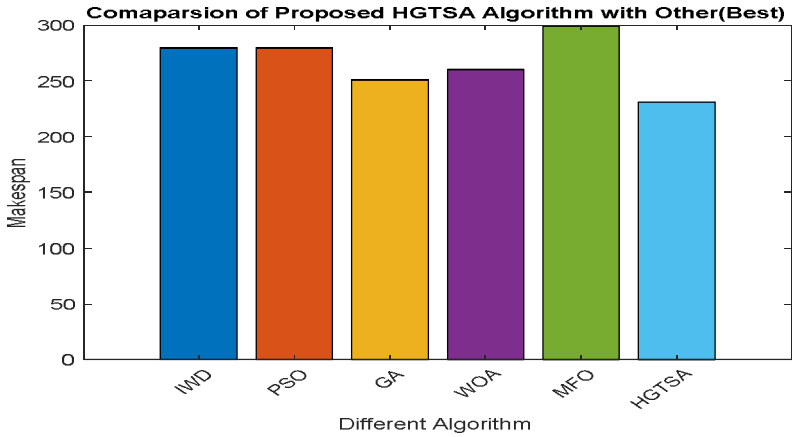


Fig. 2. Analysis & Comparison of HGTSA algorithm with other (Best) for task 20

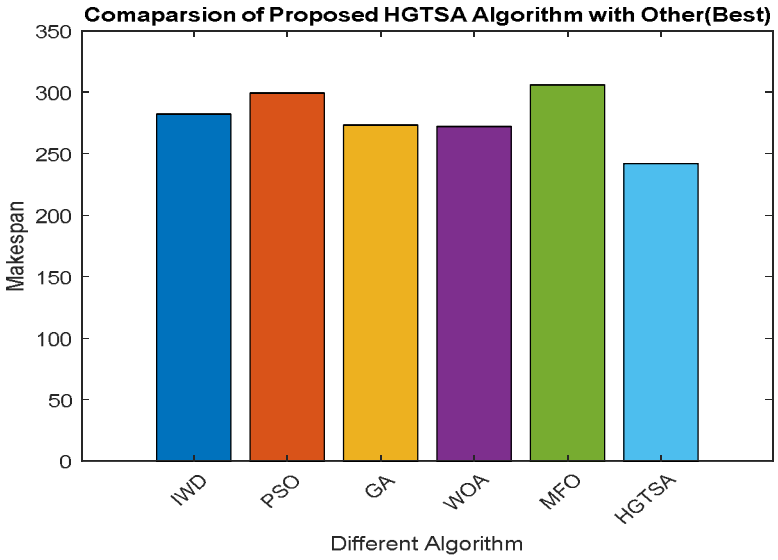


Fig3. Analysis & Comparison of HGTSA algorithm with other (Best) for task 40

Fig4 compares the makespan time of different optimization algorithms for task scheduling under best case, where task was 80. Results indicate that the MFO algorithm achieved the highest makespan time of 306, while the proposed HGTSA algorithm achieved the lowest makespan time of 257. These results demonstrate the superior performance of the HGTSA algorithm in task scheduling, and its potential to improve efficiency and effectiveness in cloud computing. Further in table 2, 3 and 4 the comparison of proposed scheme with existing approaches for 3 different cases, case 1 for 10 Tasks, case 2 for 40 Tasks and case 3 for 80n tasks are given. The comparison is made for best, medium and worst schedule solution found by proposed scheme.

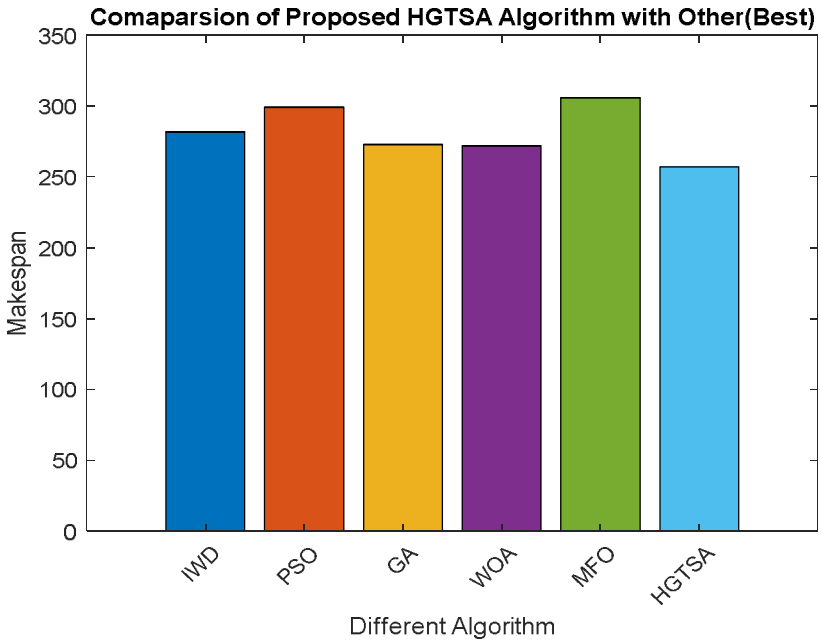


Fig 4. Analysis & Comparison of HGTSA algorithm with other (Best) for task 80

Table 2. Analysis and Comparison with Best, Medium and Worst Schedules for Case 1.

Sr. No.	Algorithms	Best	Medium	Worst
1	HGTSA	231	236	239
2	IWD [11]	280	296	301
3	GA [11]	251	256	270
4	PSO [11]	280	298	303
5	WOA [11]	260	281	290
6	MFO [11]	299	310	316

Table 3. Analysis and Comparison with Best, Medium and Worst Schedules for Case 2.

Sr. No.	Algorithm	Best	Medium	Worst
1	HGTSA	242	245	274
2	IWD [11]	282	302	306
3	PSO [11]	299	309	316
4	GA [11]	273	280	288
5	WOA [11]	272	278	291
6	MFO [11]	306	313	316

Table 4. Analysis and Comparison with Best, Medium and Worst Schedules for Case 3.

Sr. No.	Algorithms	Best	Medium	Worst
1	HGTSA	257	259	282
2	IWD [11]	282	302	306
3	PSO [11]	299	309	316
4	GA [11]	273	280	288
5	WOA [11]	272	278	291
6	MFO [11]	306	313	316

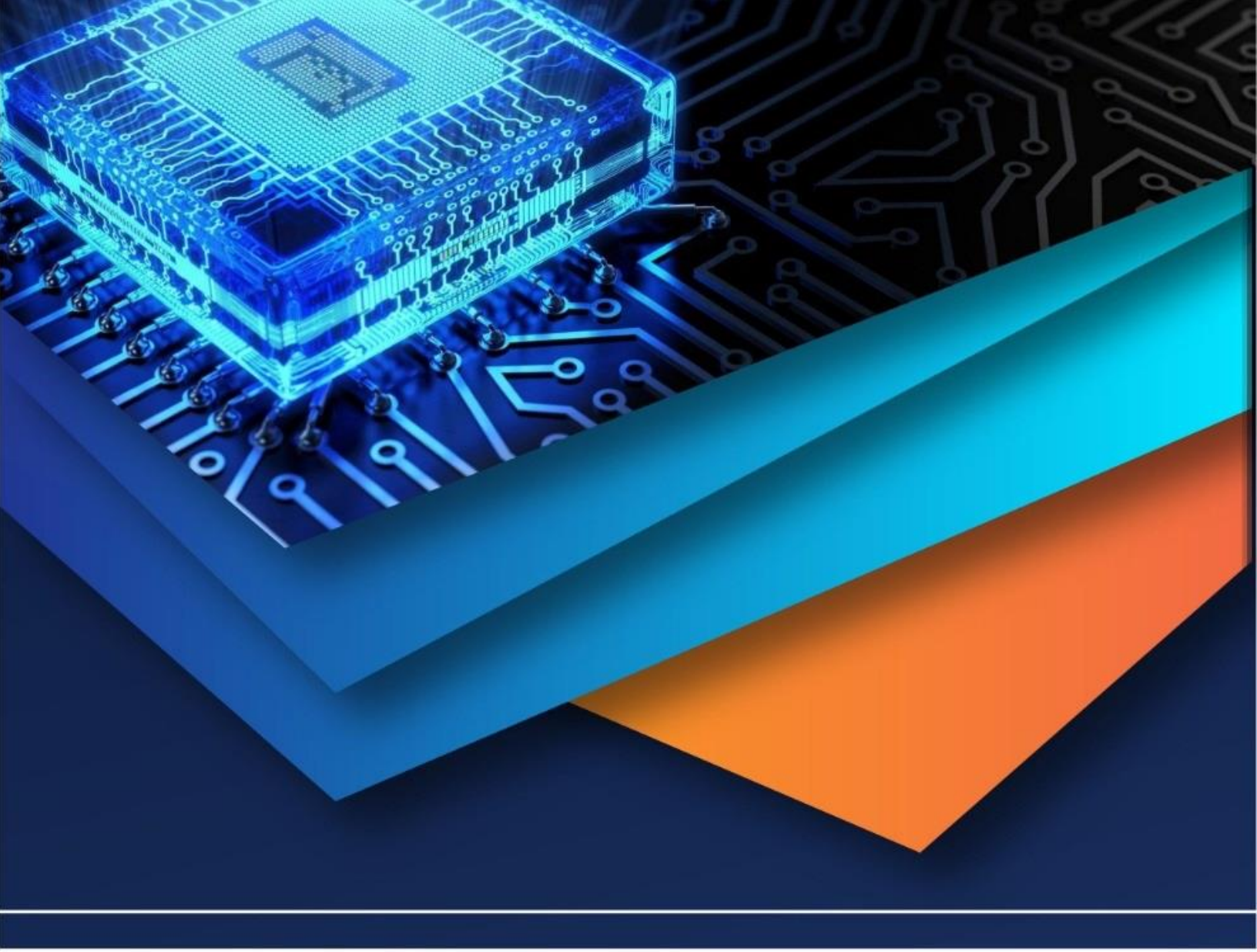
In our proposed scheme, HGTSA algorithm outperformed all five compared algorithms across the different scenarios. The results consistently demonstrated the superior performance of our algorithm, showcasing its effectiveness and competitiveness in solving the task scheduling problem. These findings highlight the significant advantages and strengths of our algorithm in comparison to the other algorithms analyzed.

VI. CONCLUSIONS

Task scheduling plays a crucial role in achieving efficient and effective utilization of resources in cloud computing systems. Our work focuses on addressing the task scheduling problem through the proposed hybrid approach, HGTSA. Compared to conventional methodologies, HGTSA offers higher convergence rates and improved overall efficiency in allocating work. The results obtained from our experiments highlight the effectiveness of the HGTSA algorithm. Under the best conditions, the algorithm achieved a superior makespan of 231, surpassing the outcomes of other algorithms. Similarly, for a task size of 40, the proposed technique performed well with a makespan time of 242. Furthermore, in scenarios with 80 tasks, the HGTSA algorithm demonstrated its efficiency by achieving a makespan time of 257. One of the key strengths of the proposed algorithm is its consideration of resource utilization and virtual machine (VM) capacity in the fitness function. By incorporating these factors, the algorithm effectively reduces system costs and enhances efficiency. These advantages make the HGTSA algorithm an efficient solution for task scheduling in cloud computing systems. In conclusion, our work contributes to the field of task scheduling by introducing the HGTSA algorithm, which outperforms conventional methodologies and achieves improved makespan times. The consideration of resource utilization and VM capacity in the fitness function enhances the efficiency and effectiveness of the algorithm. This research paves the way for more optimized task scheduling approaches in cloud computing systems, leading to improved resource utilization and overall system performance. Future research in this area can explore the application of machine learning techniques to further enhance the task scheduling process, leveraging historical data to predict workload patterns and optimize scheduling decisions.

REFERENCES

[1] S. Lipsa, R. K. Dash, N. Ivković and K. Cengiz, "Task Scheduling in Cloud Computing: A Priority-Based Heuristic Approach," in IEEE Access, vol. 11, pp. 27111-27126, 2023, doi: 10.1109/ACCESS.2023.3255781.
[2] Prity, F.S., Gazi, M.H. & Uddin, K.M.A, “A review of task scheduling in cloud computing based on nature-inspired optimization algorithm”, Cluster Comput (2023).
[3] Nabi, S.; Ahmad, M.; Ibrahim, M.; Hamam, H. “AdPSO: Adaptive PSO-Based Task Scheduling Approach for Cloud Computing”, Sensors 2022, 22, 920.
[4] Shehloo, Arif Ahmad & Butt, Muheet& Zaman, Majid. (2021), “Factors affecting cloud data-center efficiency: a scheduling algorithm-based analysis”, International Journal of Advanced Technology and Engineering Exploration. 8. 1136-1167. 10.19101/IJATEE.2021.874313.
[5] Kumar MS, Karri GR., “EEOA: Cost and Energy Efficient Task Scheduling in a Cloud-Fog Framework”, Sensors. 2023; 23(5):2445.
[6] Agarwal, M.; Srivastava, G.M.S., “Opposition-based learning inspired particle swarm optimization (OPSO) scheme for task scheduling problem in cloud computing”, J. Ambient. Intell. Humaniz. Comput. 2021, 12, 9855–9875.
[7] Malik, M.; Suman., “Lateral Wolf Based Particle Swarm Optimization (LW-PSO) for Load Balancing on Cloud Computing”., Wireless. Personal. Communication. 2022, 1, 1–20.
[8] Fortino, G.; Guerrieri, A.; Pace, P.; Savaglio, C.; Spezzano, G., “Iot platforms and security: An analysis of the leading industrial/commercial solutions”, Sensors 2022, 22, 2196.
[9] Saravanan, G., Neelakandan, S., Ezhumalai, P. et al., “Improved wild horse optimization with levy flight algorithm for effective task scheduling in cloud computing”, J Cloud Comp 12, 24 (2023).
[10] Mangalampalli S, Karri GR, Elngar AA., “An Efficient Trust-Aware Task Scheduling Algorithm in Cloud Computing Using Firefly Optimization”, Sensors. 2023; 23(3):1384.
[11] Peng, Zhihao, et al. "Genetic Algorithm-Based Task Scheduling in Cloud Computing Using MapReduce Framework." Mathematical Problems in Engineering 2022 (2022).
[12] Sanaj, M. S., and PM Joe Prathap. "An efficient approach to the map-reduce framework and genetic algorithm based whale optimization algorithm for task scheduling in cloud computing environment." Materials Today: Proceedings 37 (2021): 3199-3208.
[13] Ramezani, F., Lu, J., Taheri, J. et al., “Evolutionary algorithm-based multi-objective task scheduling optimization model in cloud environments”, World Wide Web 18, 1737–1757 (2015).
[14] Weiqing, G. E., and Cui Yanru. "Task-scheduling algorithm based on improved genetic algorithm in cloud computing environment." Recent Advances in Electrical & Electronic Engineering (Formerly Recent Patents on Electrical & Electronic Engineering) 14.1 (2021): 13-19.
[15] Nie, Jun. "Research on Task Scheduling Strategy Based on Cloud Computing Environment." Journal of Applied Science and Engineering Innovation 5.1 (2018): 9-12.
[16] Wu, Z., Xing, S., Cai, S., Xiao, Z., Ming, Z. (2017), “A Genetic-Ant-Colony Hybrid Algorithm for Task Scheduling in Cloud System”, In: Qiu, M. (eds) Smart Computing and Communication. SmartCom 2016. Lecture Notes in Computer Science(), vol 10135. Springer, Cham.
[17] Khan, M.S.A., Santhosh, R., T”ask scheduling in cloud computing using hybrid optimization algorithm”, Soft Comput 26, 13069–13079 (2022).
[18] Li, Yuxia. "ACO-SOS-based task scheduling in cloud computing." International Journal of Performability Engineering 15.9 (2019): 2534.
[19] Guo, Qiang. "Task scheduling based on ant colony optimization in cloud environment." AIP conference proceedings. Vol. 1834. No. 1. AIP Publishing LLC, 2017.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY