International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Resource Apportionment by Using Classified Krill Herd Optimization Algorithm

Saket Mishra

Submitted: 22/04/2023 **Revised**: 11/06/2023 **Accepted**: 21/06/2023

Abstract: Through the usage of the internet, cloud computing is able to provide consumers with safe, quick, and efficient services. One of cloud's most appealing features is how efficiently resources can be shared throughout applications. In cloud computing, resource allocation refers to the act of apportioning obtainable assets across users in an effective manner. Users of the cloud can rent the computing and storage resources they need from cloud service providers (CSPs), which significantly lowers their overall infrastructure costs. There are a number of different approaches to allocation in the cloud, but improving the energy efficiency of large-scale cloud data centres and optimising resource use are still significant areas of study. The computational structures determine the hierarchical assignment of resources. Even more, the krill-herd algorithm takes the calculated average values and uses them to optimise resources in a hierarchical fashion. The krill herd technique is then used to average out the hierarchical tree's resources for maximum efficiency. Experimental results demonstrate that the suggested Krill Herd methods can provide better outcomes than numerous well-known optimization strategies. The proposed method provides the optimum route to find the appropriate clustered resources for each new user request that comes in. The experimental results demonstrate that the proposed method significantly improves upon the previous method in terms of optimised execution time.

***Keywords*:** *CSP; Krill Herd Optimization; QoS; RMS.*

1. Introduction:

The phrase "cloud computing" refers to a delivery paradigm that makes use of technology that is based on the internet in order to provide service provisioning that is on-demand, scalable, and elastic. Users are also able to self-serve through the system, with allocations being determined automatically based on the amount of time and effort that individuals put in [1]. The term "Resource Allocation" (RA) refers to a mechanism that is used in the field of cloud computing to distribute accessible resources to users who have a requirement for those resources. Because there is a finite and constantly shifting pool of accessible resources in cloud computing, the most pressing task is figuring out how to allocate those resources among users in the most effective manner. Service providers are faced with a number of difficult difficulties, one of the most onerous being the optimal allocation of cloud computing resources to users.

The process of locating the resources required to complete tasks that are hosted in the cloud is an essential component of resource management [2] as shown in Figure I. This approach investigates a number of cloud resources in order to identify those that are suitable for the user, and it then distributes those resources in the appropriate manner. It helps in scheduling applications and managing resources more effectively. Yet, the finding of cloud resources is made more difficult by the centralised procedures that are used and the unpredictability of cloud resources. It is becoming increasingly difficult to predict the outcome of this scenario due to the dynamic and heterogeneous nature of clouds in environments with several clouds [3].

Associate Professor,
Centre for Interdisciplinary Research in Business and Technology,
Chitkara University Institute of Engineering and Technology,
Chitkara University, Punjab, India
saket.mishra.orp@chitkara.edu.in

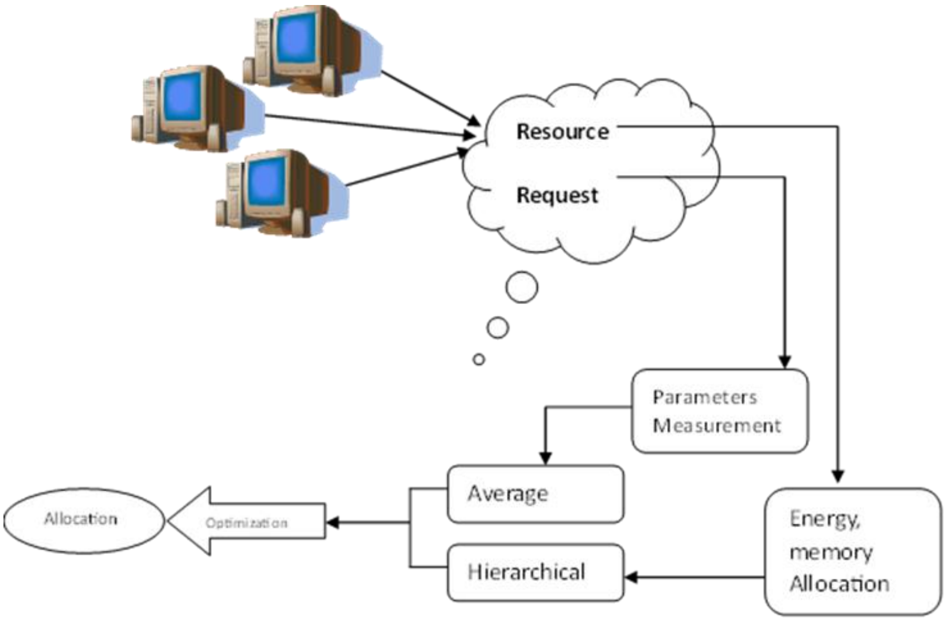

Fig I: Architecture of Resource Allocation Model

Questions such as "how can one locate resources with closer proximity in a dispersed cloud?" and "how can one minimally influence the network, particularly expensive inter domain traffic?" are frequently asked in the context of resource discovery. For example, "how can one locate resources with closer proximity in a cloud?" The difficulties at question have, quite simply, quite obvious remedies. When dealing with NV-related circumstances, resource discovery services frequently make use of a discovery structure that incorporates an advertisement process. It is utilised by Brokers in order to gain knowledge about and evaluate the accessible resource offers made available by various Cloud Service Providers [4]. The resource descriptors and resource states that are archived are the responsibility of the shared repositories that it incorporates. Taking into consideration that one of the most important aspects of cloud computing is the flexibility to dynamically alter the amount of resources that are allocated.

Automated resource provisioning is highly dependent on resource monitoring; as a result, resource monitoring is an essential component for maximising resource utilisation. Its active or passive nature can be inferred from the circumstances surrounding it. The action is considered to be passive when information is being obtained by a third party or parties from the outside [5]. The organisation has the option to either conduct this action on an ongoing basis by sending polling email requests to systems for data or on an as-needed basis. As long as systems are self-sufficient, resource monitoring will be carried out, and each system will have the autonomy to choose when to provide asynchronous state data to the relevant authorities. Both approaches can be utilised on a continuous basis in shared clouds to further improve the degree of resource tracking precision achieved. It is vital to synchronise updates across repositories if one wishes for the data pertaining to the state to be constant and dependable [6].

In order to satisfy user requirements, the system keeps track of the resources that are available and selects the ones that are the best fit. In addition to this, it helps you make the most of the resources at your disposal. Finding resources that are suitable for the intended application as well as the request should be the goal of any method that may be used to choose those resources.

It is not an easy task to select the appropriate resource from a pool because of characteristics such as shifting resource availability, high algorithmic complexity, and any other factors that may be pertinent to cloud service providers [7]. It is usual practise to employ optimization algorithms when selecting the most appropriate resources.

The process of optimization makes use of a wide variety of strategies, ranging from fundamental heuristics to well-established metaheuristic algorithms. The distribution of available resources can also be made more efficient by the application of artificial intelligence algorithms, such as those derived from Game Theory and biology [8].

There are two different ways to make decisions on resource strategy: a priori and a posteriori. In the a priori scenario, the solution that was suggested initially for the assignment was the best one that could possibly be proposed. In order to successfully achieve this objective, the policy would take into account all of the pertinent criteria that are related with the assignment. So to start, we are given a choice that is not nearly as good as we would like. It is necessary for the Cloud Service Provider to practise consistent resource management in order to improve the quality of this service over the long term. It may be necessary to make choices like allocating and reallocating resources in order to maximise the utilisation of available resources and satisfy the requirements of cloud users [9].

Because the usage of resources and the provisioning of them are both powerful and time-dependent processes, it is essential that any technique for posteriori optimization arrives to the optimal assignment step in a short amount of time. In addition, it is essential to maximise the use of existing solutions while persistently working to improve them to accommodate shifting customer requirements.

The discovery process is consequently essential to the resource management discipline as it serves as the basis for effective resource allocation and management, both of which are heavily reliant on its completion. It searches for the most useful resource there is so that it can satisfy the requirements of the application. The majority of the time, this is something that is handled by the Cloud Service Provider [10]. This mechanism for searching for and locating the required resources in the system was conceived of and developed by the user agent or resource broker. Because of resource discovery, cloud resource management systems are able to communicate with one another as well as with other RMSs that are operating within the same cloud environment. In addition, the process of discovery includes the step of updating the server with the most recent information regarding the resources.

2. Existing Work Done:

The challenge of optimal resource allocation in the presence of time-varying workloads and undefined channels has been investigated by the authors, who have considered a virtual data centre (calculating cloud) consisting of a group of servers presenting several mobile terminals (i.e., a mobile cloud). In PSP and SSP cognitive networks, the ambiguity of the channel could have resulted from fading or from an infinitely available link and consistency. The goal of the planned optimal resource allocation policy was to maintain network stability while increasing the cooperative efficiency of the long-term average throughput and decreasing the combined energy consumption at terminuses and servers [11].

In-depth analyses of various strategies for allocating cloud resources have been conducted by researchers. The section on findings and analysis detailed the merits and procedures of each approach. Resource allocation in the cloud was analysed, and the paper's conclusion was that there was no optimal method [12].

A scalable hybrid Cloud architecture and resource provisioning processes have been presented by the researchers to guarantee the QoS goals of the users. The desired policies take into account the workload replica and failure correlations in order to reroute user demand to the appropriate Cloud providers [13]. The performance, cost, and efficiency of performance to cost of the planned resource provisioning rules were tested using the actual failure traces and workload model. The results of the simulation showed that in a realistic working environment, where the user estimates for the demands were incorporated into the provisioning procedures, the QoS of the user was improved by nearly 32% on the basis of deadline violation rate and 57% on the basis of slowdown with constrained cost on public Cloud [14].

Researchers have provided a dynamic control technique that boosts the scheme's long-term average throughput by adaptively allocating resources over time-variable fading channels. As an added bonus, the concept of Inter System Networking (InSyNet) was introduced to provide a performance bound for the allocation resource techniques of the Cooperative Communication Network (CCN) [15].

Writers have considered how to approach the allocation of resources and pricing strategies in a Compute Cloud for both autonomous tasks and workflow system jobs. Workflow task scheduling was difficult since different parts of the workflow can need more resources, which might slow down the whole thing. We used The Nash Bargaining Solution (NBS) and another axiomatic bargaining approach. The goal of implementing the Raiffa Bargaining Solution (RBS) was to find the best method of allocating Compute Cloud's virtual CPU cores for both unattended and scheduled jobs [16]. Simulation experiments examined the methods' efficacy and showed that they could be modified to suit the needs of Cloud service providers (CSPs) for estimating resource requirements. It was also discovered that the NBS ensures proportional fairness while the RBS can handle incoming tasks and their dynamics in real time [17].

In this context of Cloud Computing, researchers have focused on the problem of resource allocation. In most cases, this task will involve the issue of improving the cost mapping of Infrastructure as Cloud Service (IaaS) on a Network Edge Data-Centers (DCs) with regards to Quality of Service (QoS) requirements. To deal with IaaS needs, the paper recommended energically segmenting data centre network resources into distinct QoS categories [18].

Many attempts have been made to map IaaS in the literature, but they have largely ignored the QoS dynamics in favour of the cloud hosting requirements. In the end, they were unable to provide Quality of Service guarantees for standard IaaS requirements, leading to a high level of customer discontent [19].

The speed of a network mostly depends on its bandwidth. The term "bandwidth" describes the maximum data transfer rate of a network connection in the context of computer networking. It is a measure of the connection's potential throughput in terms of data bytes per second. Larger storage capacities almost always lead to increased efficiency [20]. In practise, bandwidth and throughput are lower than their theoretical maximums. The following illustration clarifies the difference. Theoretical throughput for a basic 802.11g Wi-Fi connection is 54 Mbps, but in reality, users should expect to receive throughput of just about 25 Mbps [21-23].

Theoretically, the maximum throughput of a standard Ethernet network is either 100 Mbps or 1000 Mbps. Therefore, it is unrealistic to expect to ever actually attain this maximum. Networks that use the same idea, such as cellular (mobile), networks, typically do not advertise their bandwidth speeds [24-25]. Difference between potential bandwidth and actual throughput is driven by communications overheads in computer hardware, network protocols, and operating systems.

3. The Proposed Work:

In this paper study about the unique bioinspired optimization method Krill Herd (KH) algorithm as shown in Figure II and its application to cloud resource allocation is done. Possibly it takes into account the energy used by each resource before assigning them. For large-scale scientific workflow applications, this method is more beneficial to increase resource utilisation. Taking cues from the krill's herding behaviour, we provide the KH method for resolving optimization issues. Idolizing the mobility features of the krill individuals allows for the creation of a wide variety of algorithms. An individual's position in time is determined by three fundamental actions:

- Actions prompted by the presence of other people,
- Hunting and gathering, and
- Random Dispersion

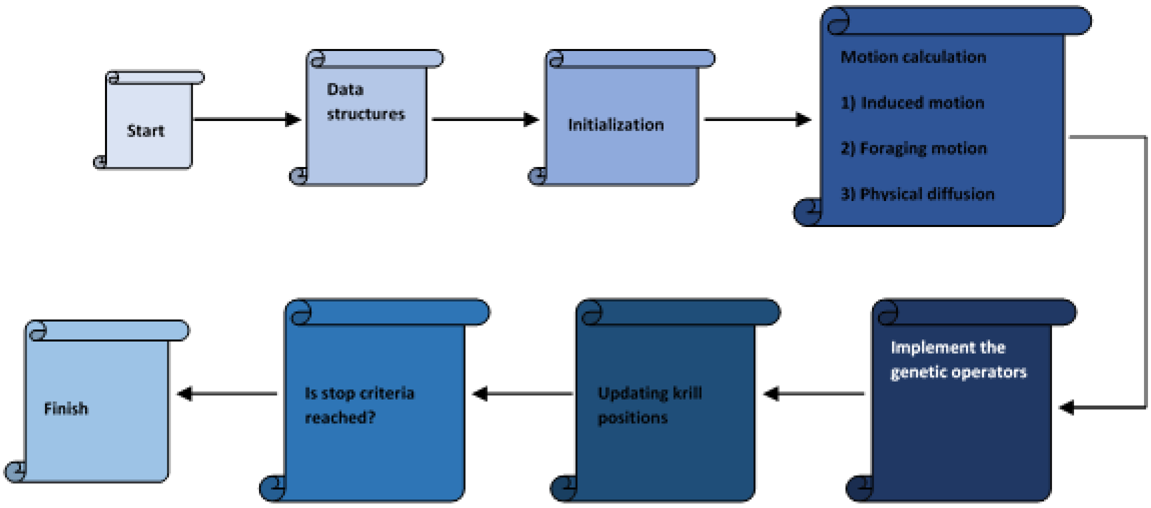

Fig II: Proposed Block Diagram.

For the most part, the KH algorithm can be presented in the following ways:

Step 1: We specify the simple boundaries and initialise the algorithm's parameters.

Step 2: Generate a random sample of the search space's initial population

Step 3: we rank each item based on how valuable it is.

Step 4: Socially-Induced Foraging Movement According to the laws of physics, diffusion

Step 5: Use Genetic Operators to Probe the Search Area

Step 6: Recalculating where each krill is in the search space.

Step 7: is to continue on to Step III until the Ending Condition is met.

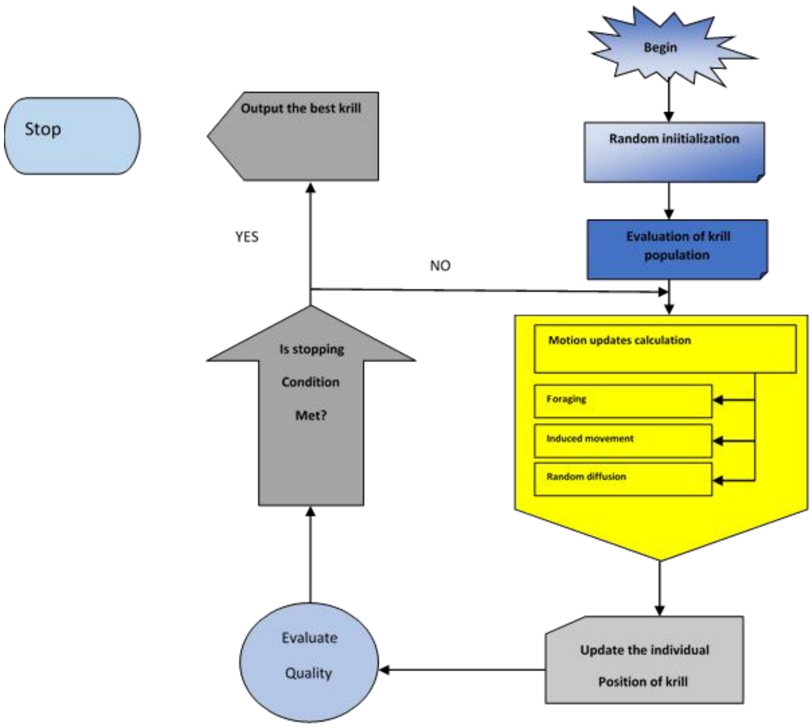

Fig III: Proposed flow Diagram.

The proposed algorithm as shown in Figure III is intriguing since it leverages real-world empirical investigations to obtain the coefficients and carefully simulates the behaviour of krill. This means that the only parameter in the KH algorithm (Table I) that needs tuning is the time interval. As compared to similar nature-inspired algorithms, this may be seen as a significant benefit of the suggested approach. However, for optimal performance, the KH algorithm's parameters should be fine-tuned using an effective meta-optimization technique tailored to the specific situation at hand.

Table I: Pseudo code for the krill herd algorithm.

```
Begin
(i) Size (S) of the population and iteration (Imax)
(ii) to have a completely random starting point.
Initialize the population
Set the iteration counter to 1.
Adjust the maximum foraging velocity, maximum search depth, and maximum search number.
(iii). Assessment of Physical Health.
To determine an accurate population estimate, we must first determine the size of the krill population.
(iv) But I<Imax does rank the krill from best to worst.
Do the following motion calculation for x = 1: S do.
Changes in direction prompted by congregating krill, Physical dispersion, Foraging Activities
Refresh each krill's location in the search matrix.
Stop at x
The best krill can be found by ranking the others from best to worst.
Imax = I+1.
Finally, estimate the krill optimal solution and exit the while loop.
End
```

4. Result and Analysis:

Using Java and Cloud Sim, we measure the execution time, throughput, and time-based task performance of the proposed Krill Herd based resource allocation. All the tests are performed on a Linux Windows 10 PC that has a 3.4 GHz Intel processor and 4 GB of main memory.

4.1. Tasks can be triggered to run at specific times or in response to specific events, a process known as "time-based task execution". Time required for the Krill Herd algorithm to complete a task is compared to that of the standard Grouped tasks scheduling (GTS) algorithm. A visual comparison of the execution times of our suggested krill herd and the GTS algorithm is presented in Figure IV.

Table II: Execution Time comparison with Existing Approach.

S. No.	Algorithm	Execution Time
1	HABCCS	14800
2	GTS Algorithm	13000
3	Krill	9000

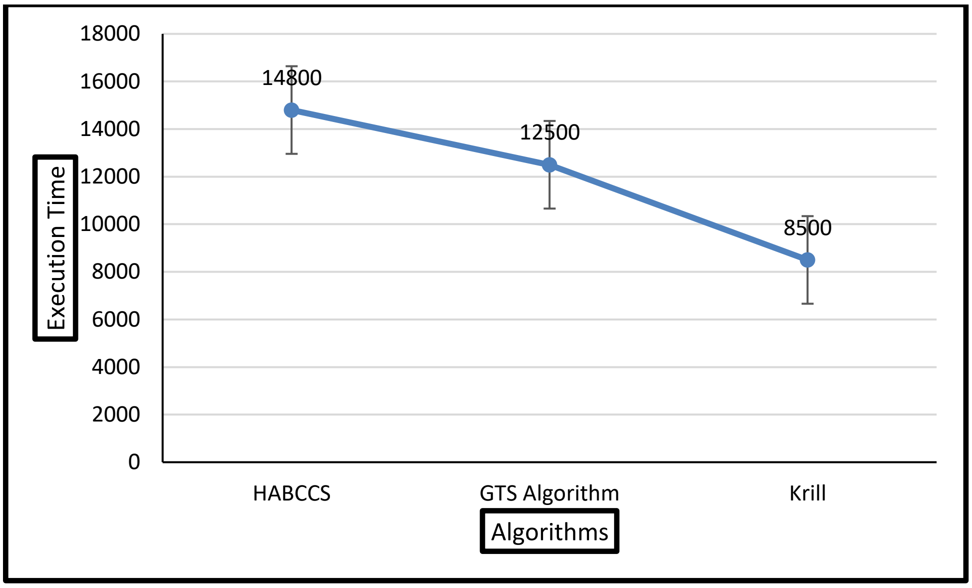

Fig IV: Execution Time comparison with Existing Approach.

4.2. The throughput is the amount of data that can be transferred from one location to another in a given amount of time, and it is expressed in bits per second (bps). Simply explained, throughput is the highest possible production rate or the fastest possible processing speed. The suggested system has a higher throughput than the existing systems.

Table III: Throughput comparison with Existing Approach.

S. No.	Algorithm	Execution Time
1	HABCCS	18
2	GTS Algorithm	36
3	Krill	54

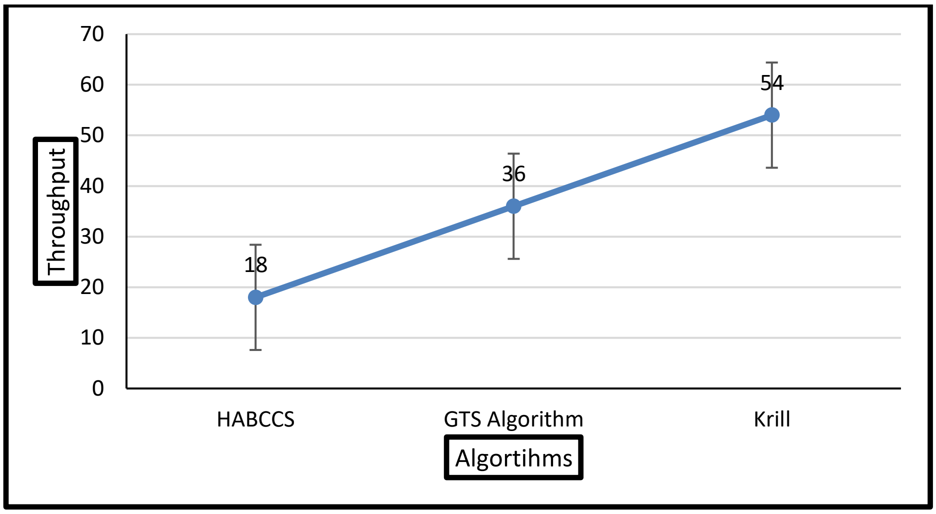

Fig V: Throughput comparison with Existing Approach.

4.3. It was the total amount of time between when a task was submitted and when it was finished that was spent on its execution. It controls how long the programme takes to reply to the user. The latency measures how long it takes from when a process makes a request until that request is carried out. This can be done with the help of the following equation:

$$R(t) = T(t) - A(t) \quad (1)$$

R(t): Turn Round Time

T(t): Task Accomplishment time

A(t): Average Appearance time

Table IV: Turn Round Time comparison with Existing Approach.

S. No	Task	GTS Algorithm	Krill Algorithm
1	50	40	0
2	150	55	25
3	200	300	275
4	250	180	180

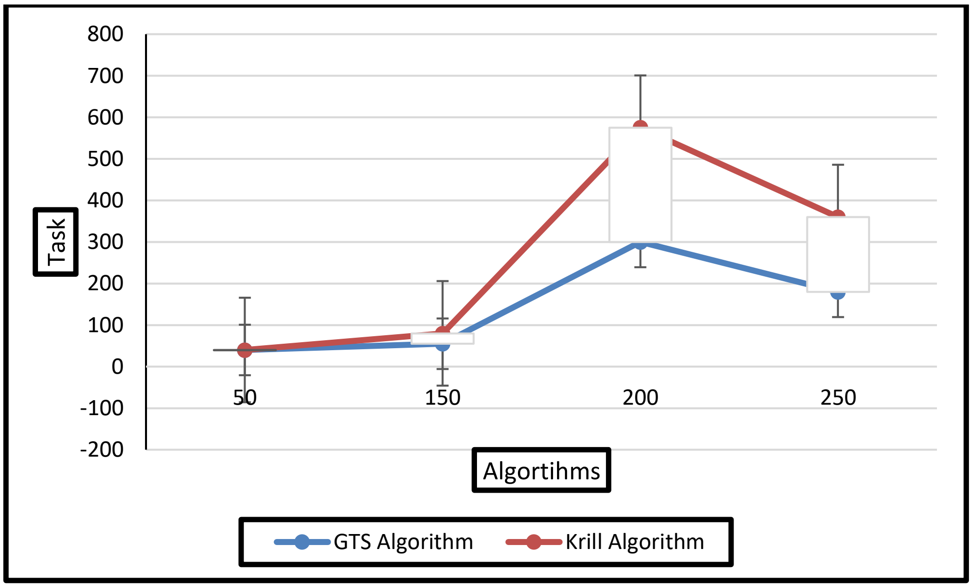

Fig VI: Turn Round Time comparison with Existing Approach.

Average Waiting Time:

It measures the typical lag time between when a task is requested and when it is completed. Below is an equation that may be used to determine the mean time,

$$T_{wait} = R(t)_{avg} - T_{\text{Spurt}} \quad (2)$$

Where, T_{wait}= Average waiting Time;

$R(t)_{avg}$= Average Turn-around Time;

T_{Spurt}= Spurt Time.

Table V: Average Waiting Time comparison with Existing Approach.

S. No	Task	GTS Algorithm	Krill Algorithm
1	50	4	1
2	150	3	0.15
3	200	2	0.28
4	250	0.89	0.060

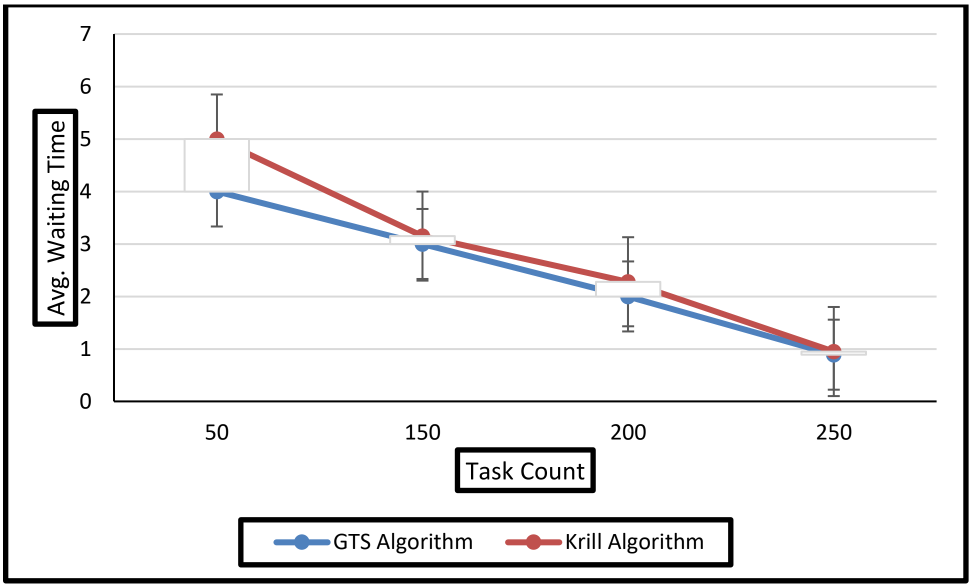

Fig VII: Average Waiting Time comparison with Existing Approach.

The krill approach typically has a shorter average waiting time than the GTS one. The analysis presented here demonstrates how the Krill Herd Algorithm may be applied to efficiently distribute cloud assets among requests. The effectiveness of the algorithm is evaluated virtually. The experimental evidence shows that the suggested algorithm outperforms the state-of-the-art algorithms on a number of metrics, such as response and waiting times.

5. Conclusion:

The study's ultimate goal is to enhance cloud computing's already impressive resource management. Due of the cloud's centralised resource management, it can be challenging to quickly assign the necessary resources to user requests. This paper's primary focus is on developing better methods for allocating cloud computing resources to applications, such as large-scale scientific operations. In this paper, we offer method for allocating resources that make use of cutting-edge hybrid optimization techniques.

The estimated average values are used by the Krill-Herd algorithm to optimise resources in a hierarchical fashion. We simulate a large number of queries to see how well the proposed method scales. The experimental results validate that the suggested Krill Herd makes efficient use of available resources. The throughput and the execution time are also enhanced. The findings validate the validity of the proposed strategy as a means to optimally allocate resources.

Although this study enhances the effectiveness of resource allocation, multi-objective meta-heuristic algorithms can further enhance resource management. Expanding on the methods provided in this paper requires adopting a more nuanced hybridization strategy for allocating resources to tasks. The effectiveness of the proposed methods can eventually be measured in real time. In addition, researchers can use alterations to meta-heuristic methods to better distribute cloud resources. Tasks with various needs can be handled in multiple ways, which improves resource allocation.

Conflict of Interests:

The authors declare that there is no conflict of interests regarding the publication of this paper.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References:

[1] Zhang, Jiangtao, Hejiao Huang, & Xuan Wang, 2016, ‘Resource provision algorithms in cloud computing: A survey’, Journal of Network and Computer Applications, vol. 64, pp. 23-42.

[2] Wei, & Wei, 2016, ‘Imperfect Information Dynamic Stackelberg Game Based Resource Allocation Using Hidden Markov for Cloud Computing’, IEEE transactions on services computing, pp. 1-13.

[3] Wu Mingxin, 2015,‘Research on Improvement of Task Scheduling Algorithm in Cloud Computing’, Applied Mathematics & Information Sciences ,vol. 9, no. 1, pp. 507-516.

[4] Vikas Kumar, & Ankur Singh Bist, 2015, ‘Cluster Analysis: A Survey’, International Journal of Engineering Sciences & Research Technology, vol. 4, no. 3.

[5] Wagh, & SupriyaSantosh, 2015, ‘Prospect Of Adopting Cloud Computing In Indian Manufacturing Industry’, International Journal of management, vol. 6, no. 6.

[6] Tso, Fung Po, Simon Jouet, & Dimitrios P. Pezaros, 2016, ‘Network and server resource management strategies for data centre infrastructures: A survey.’ ,Computer Networks, vol. 106 , pp.209-225.

[7] Vakilinia, Shahin, Mohamed Cheriet, & JananjoyRajkumar, 2016, ‘Dynamic resource allocation of smart home workloads in the cloud.’ ,Proceedings of 12th International Conference on Network and Service Management , pp. 367-370.

[8] Sukhpal Singh, & Inderveer Chana, 2016, ‘A survey on resource scheduling in cloud computing: Issues and challenges’, Journal of grid computing, vol. 14, no. 2, pp. 217-264.

[9] Syed Hamid Hussain Madni, Muhammad Shafie Abd Latiff, & Yahaya Coulibaly, 2017, ‘Recent advancements in resource allocation techniques for cloud computing environment: a systematic review’, Cluster Computing ,vol. 20, no. 3, pp. 2489-2533.

[10] Shu, Wanneng, Wei Wang, & Yunji Wang,2014, ‘A novel energy efficient resource allocation algorithm based on immune clonal optimization for green cloud computing’, EURASIP Journal on Wireless Communications and Networking, vol. 64, no. 1, pp. 0-8

[11] Singh, Roshni, & Shiva Prakash, 2016, ‘Survey on Dynamic Resource Allocation Schemes in Cloud Environment: A Review’, Journal of Network and Computer Applications, vol. 2, no 3.

[12] Seyedehmehrnaz Mireslami, Logan Rakai, Behrouz Homayoun Far, & Mea Wang, 2017, ‘Simultaneous Cost and QoS Optimization for Cloud Resource Allocation’, IEEE Transactions on Network and Service Management, vol. 14, no. 3, pp. 676-689.

[13] Sharma, & Yogesh, 2016, ‘Reliability and energy efficiency in cloud computing systems: Survey and taxonomy’, Journal of Network and Computer Applications, vol. 74, pp. 66-85.

[14] M. Hamzeh, B. Vahidi, and A. F. Nematollahi, “Optimizing configuration of cyber network considering graph theory structure and teaching–learning-based optimization (gttlbo),” IEEE Transactions on Industrial Informatics, vol. 15, no. 4, pp. 2083–2090, 2019.

[15] C. Zhang, H. Xu, and Y. Li, “Spatial–spectral semi supervised classification based on teaching–learning-based optimization for hyperspectral image,” Journal of Applied Remote Sensing, vol. 12, no. 02, p. 1, 2018.

[16] G. Sharma and A. Kumar, “Modified energy-efficient range free localization using teaching-learning-based optimization for wireless sensor networks,” IETE Journal of Research, vol. 64, no. 1, pp. 124–138, 2018.

[17] X. J. Bi and T. W. Pan, “Relevance feedback image retrieval based on teaching-learning-based optimization algorithm,” Tien Tzu Hsueh Pao/Acta Electronica Sinica, vol. 45, no. 7, pp. 1668–1676, 2017.

[18] S. D. K. Ram, S. Srivastava, and K. K. Mishra, “A Multi Objective Generalized Teacher-Learning-Based-Optimization Algorithm,” Journal of 5e Institution of Engineers (India), pp. 1–16, 2022.

[19] Y. Hassanzadeh, A. Jafari-Bavil-Olyaei, M. T. Aalami, and N. Kardan, “Experimental and numerical investigation of bridge pier scour estimation using anfis and teaching–learning-based optimization methods,” Engineering with Computers, vol. 35, no. 3, pp. 1103–1120, 2019.

[20] Yang, W.; Xia, K.; Fan, S.; Wang, L.; Li, T.; Zhang, J.; Feng, Y. A multi-strategy Whale optimization algorithm and its application. Eng. Appl. Artif. Intell. 2022, 108, 104558.

[21] Liu, L.; Zhang, R. Multi-strategy Improved Whale Optimization Algorithm and Its Application.

Computational Intelligence, Neuroscience, 2022, 2022.
[22] Altay, E.V.; Alatas, B. Bird swarm algorithms with chaotic mapping. Artificial Intelligence, Rev. 2020, 53, 1373–1414.
[23] Shirvani, M.H. Bi-objective web service composition problem in multi-cloud environment: A bi-objective time-varying particle swarm optimisation algorithm. J. Exp. Theor. Artif. Intell. 2021, 33, 179–202.
[24] Seghir, F.; Khababa, A.; Semchedine, F. An interval-based multi-objective artificial bee colony algorithm for solving the web service composition under uncertain QoS. J. Supercomput. 2019, 75, 5622–5666.
[25] Arunachalam, N.; Amuthan, A. Integrated probability multi-search and solution acceptance rule-based artificial bee colony optimization scheme for web service composition. Nat. Comput. 2021, 20, 23–38.
[26] Rajan, S. ., & Joseph, L. . (2023). An Adaptable Optimal Network Topology Model for Efficient Data Centre Design in Storage Area Networks. International Journal on Recent and Innovation Trends in Computing and Communication, 11(2s), 43–50. https://doi.org/10.17762/ijritcc.v11i2s.6027
[27] Dhabliya, D. (2021). Feature Selection Intrusion Detection System for The Attack Classification with Data Summarization. Machine Learning Applications in Engineering Education and Management, 1(1), 20–25. Retrieved from http://yashikajournals.com/index.php/mlaeem/article/view/8