ADAPTIVE NEURO-FUZZY INFERENCE-QOS AWARE GENETIC ALGORITHM FOR RESOURCE ALLOCATION OPTIMIZATION IN CLOUD

D. Komalavalli

Research Scholar, Bharathiar University,
Coimbatore, Tamilnadu, India
komsdayalan@gmail.com

T. Padma

Professor, Sona College of Technology,
Salem, Tamilnadu, India
padmatheagarajan@gmail.com

Abstract

Cloud computing, a new and promising distributed computing technology, offers a pay-per-use foundation for large-scale scientific and business process applications. With the advancement of cloud computing, artificial intelligence, and big data, demands for cloud resources, particularly certain unclear and emergent resource demands, are fast increasing. Classical cloud resource allocation approaches do not assist the emergent mode in terms of ensuring resource allocation timeliness and optimization. The proposed method employs an Adaptive Neuro-Fuzzy Inference System Quality of Service Aware Genetic Algorithm (ANFIS-QoSGA) method for quickly determining the best VM for each job. This paper proposed an improved resource allocation optimization technique that takes into account the goals of reducing deployment costs, balancing the load, and increasing the Quality of Service performance. Cloud customers' main challenge is deciding the resources to use for the deployment of their applications without negotiating the Quality of Service requirements. The proposed algorithm takes into account 'n' Number of customers Quality of Service needs and resources allocated inside the limited cost constraints. The proposed approach's major goal is to reduce task computational time, cost, and energy consumption while maximizing meaningful resource use. Extensive studies show that the suggested method outperforms other similar scheduling techniques in terms of energy cost and has a better outcome in terms of total execution time which is reduced by 4%, 8%, and 11% when compared with RAA-PI-NSGAII, SFWOA,NSGA-III and makespan, degree of imbalance, and security value under high load conditions.

Keywords: **Cloud Computing; Resource Allocation; ANFIS; VM; QoS; GA; deployment cost.**

1. Introduction

The term "cloud computing"[1,2] refers to a collection of resources including computational servers, storage servers, and database servers. Cloud offers clients these services on a pay-as-you-go basis. Virtualization technology is used in cloud computing to partition enormous physical resources into multiple virtual resources. In a cloud platform, a huge number of clients can access these virtual resources at any time and from any location. With the implementation and advance of cloud computing, artificial intelligence and big data, an increasing number of applications are being installed on cloud platforms, whose resource demands have become progressively huge. In recent decades, the academic and industrial communities have been mutually more interested in the cloud paradigm. Instead of retaining and establishing a local data center, most people and company systems suggested to delegate enormous amounts of data to a cloud database. Cloud customers have access to a variety of cloud services provided via the public cloud. The cloud computing model, on the other hand, is a new design in the cloud setting that permits for the efficient use of resources, infrastructure, and energy at 1 or more levels of abstraction, with services delivered over a computer network or the internet.

In the cloud, resource distribution based on customer application requirements is a big challenge. Determining the appropriate physical servers on which to install virtual machines (VMs) is a challenge of VM placement in the resource allocation process [3-5]. Resource allocation is utilized in the cloud framework to improve customer happiness while reducing processing time. Minimizing resource usage ensures cloud service quality, satisfaction for the service provider, and increased throughput. This technique can not only meet VM resource demands, but also increase the cloud platform's resource consumption. A variety of algorithms [6] are presented to resolve the resource allocation problem. Resource allocation is a major issue in cloud data centers, and it is frequently described as a single objective function. To formulate the challenge, we used the proposed model's QoS performance as the main objective. The key objectives for issue formulation in several models are QoS, energy usage and deployment cost. For discovering discrete solutions, the suggested technique employs a branch and bound algorithm.

The following are the paper's main contributions:

- The goals of the proposed method are to reduce deployment costs, balance the load and increase QoS performance.
- The ANFIS model takes input parameters such as cost, energy, resource utilization, and QoS requirements to achieve the best outcomes.
- These output parameters are fine-tuned utilizing the genetic algorithm (GA).
- The GA algorithm aids ANFIS in adjusting the membership functions. The algorithm's result is ensured to be the prime optimal solution.
- Consider primary scheduling objectives such as computing time, energy consumption and resource usage while designing the multi-objective optimization problem.
- The rest of the paper is laid out as follows. Section 2 discusses the research that has been done in relation to the present resource allocation strategies. Section 3 shows how to formulate an issue as well as how to allocate resources. The optimization approach for cloud resource allocation is discussed in Section 4. The performance evaluation with respect to various workloads is explained in Section 5. Finally, Sect. 6 brings the study to a conclusion.

2. Related Work

This section examines a variety of existing allocation strategies. Subhash et al. [7] presented the Sunflower Whale Optimization Approach (SFWOA) as an effective optimization algorithm for resolving resource allocation difficulties. To obtain optimal resource allocation, it exploited the humpback whale's hunting approach and foraging behaviour, as well as the unusual behaviour of sunflowers. By adjusting the number of iterations, SFWOA was able to achieve the best utilization of resources of 0.942 utilizing 20 VMs, maximum utilization of memory of 0.215, maximum utilization of CPU of 0.269 utilizing 15 VMs, and minimal skewness of 0.001 with 25 VM's. Gong et al. [8] suggested an adaptive control strategy for allocating cloud resources based on resource demands and workload requests. Using multivariable control, numerous resources were successfully distributed to multiple services based on varying requests. This strategy addressed the service need and enhanced resource consumption, however it didn't assess efficiency.

Chen et al. [9] proposed the RAA-PI-NSGAII evolutionary method for solving the multi-objective optimization paradigm, that not only enhances the features and distribution consistency of the set of solutions but it also speeds up the resolving time. The outcomes of the experiments revealed that the algorithm is capable of not just allocating resources swiftly and ideally for emerging demands, but can also balance the use of all types of resources. To improve cloud energy efficiency, Li-Der Chou et al. [10] created a dynamic power saving resource allocation (DPRA) employing the PSO technique. The least-square regression model was utilized to forecast PM resource use. To improve the implementation of the data center in the cloud, it neglected to account the progress iteration and the time of selection. It could be used for topology structures with a variety of networking capabilities.

Xi Liu et al. [11] proposed a swarm optimization approach for more efficient cloud resource allocation. By balancing the exploitation and exploration parameters, it successfully used the searching behavior and improved the algorithm's performance. This strategy maximized resource utilization while neglecting to account for scalability.

To improve both agreement index and resilience, Miriam et al. [12] suggested a novel resource allocation methodology for cloud service providers to plan and allocate resources efficiently. The resource allocation scheme's main idea is to use the Non-Dominated Sorting Genetic Algorithm (NSGA-III) to efficiently distribute resources. In addition, the NSGA-III was updated to accommodate any interim data sources. Furthermore, because the search is not based on any indices, it is faster than other optimization techniques. Wang et al. [13] proposed a decentralized multi-agent (MA) supported allocation of VM mechanism in the cloud. The allocation of VM to each PM was decided using an auction-based methodology. The negotiation-based paradigm was created to swap duties in order to save energy costs. It failed to handle the bi-objectives as well as the energy cost at the same time.

3. Problem Formulation

The main problem with cloud computing is that choosing the right combination of resources depending on the user's needs [14] is a difficult undertaking. Most users supply several activities with various requirements, like computing time, energy usage, memory use, time delay, transmission of data, reaction time, and so on, and they are referred to as the complexity of modeling and cost, energy, total cost, and so on. When handling an greater number of user requests in cloud computing, start-up time, scheduling cost, execution time, computation, and energy are all critical for scheduling algorithms. Another difficulty with scheduling is that it necessitates more time spent in the environment. As a result, optimization methods can be utilized in the cloud computing environment to reduce the computational assessing job scheduling. The problem is structured for the most efficient use of resources in cloud, and it uses data from cloud service providers' resources as input. Although the service providers supply customers with similar resources, they differ with regard to price range, QoS performance, and service kind. All information regarding the resources needed by the clients is displayed by the service providers.

3.1. *System Model*

The cloud data center paradigm is represented in Fig. 1 by a group of computing servers that can accept containers and virtual machine (VM) events depending on the job requirements. The cloud environment is made up of the numbers of service providers who supply services and the infrastructures of consumers who make requests. When a large number of requests for identical resources arrive in the cloud, there is a resource shortage, and the question is how to allocate the resources to the users. For task execution, each virtual machine obtains numerous configurations, such as memory, cost, and size.

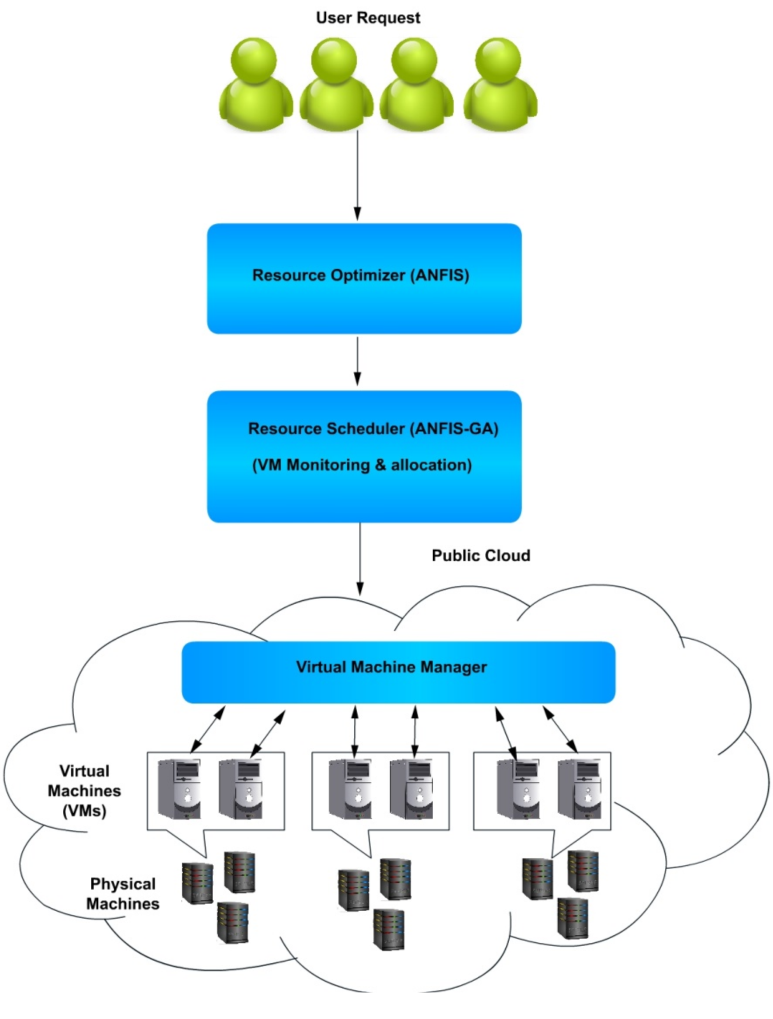


Fig 1: Proposed Framework

Customers' needs, such as QoS and money to install the application, are also included in this problem description. The customer requirements are shown in Table 1.The deployment of user applications in the cloud necessitates a variety of resources at various levels. As a result, the customer must buy the resources needed to meet the application's requirements.

Notation	Description
T_d	Total duration for application distribution
E_{cd}	Predicted computing demand
E_{dbd}	Expected demand for database
Rt_{db}	Time of response of instance for database
Rt_c	Time of response for computing instance
Rq_s	Application Storage required
C_{max}	Maximum Cost for application deployment

Table 1: Customer Requirements

3.2. *Load balancing*

In cloud computing, load balancing ensures that resources such as hard drives (storage), virtual machines (VMs), network interfaces, and servers deployed on physical servers are distributed equally. When loads are scheduled on VMs running on physical nodes, it is possible that some of the VMs will be overused while others would be underused. As virtual machines (VMs) become overused, their makespan (the time it takes for a VM to finish all the tasks assigned to it) rises. If VMs are under-utilized, the makespan falls while the cost of resource usage increases since the available resources (VMs) are not fully employed, that result in resource wastage. Reduced makespan must not be accompanied by higher resource consumption costs, and vice versa.

The concept of load balancing relates to the procedure of reallocating workload in a distributed system, so that no server becomes overused, underused, or idle. By accelerating several limited characteristics such as response time, execution time, system stability, and so on, load balancing improves cloud performance. This is an NP-hard optimization technique for scheduling jobs. The method to improve the quality of services (QoS) is to balance the load among machines, which will assist in increasing resource usage and throughput, while minimizing the makespan, response time , latency, and cost of balancing the load across machines. Therefore, the load should be distributed among the VMs so that both makespan and costs can be controlled and balanced.

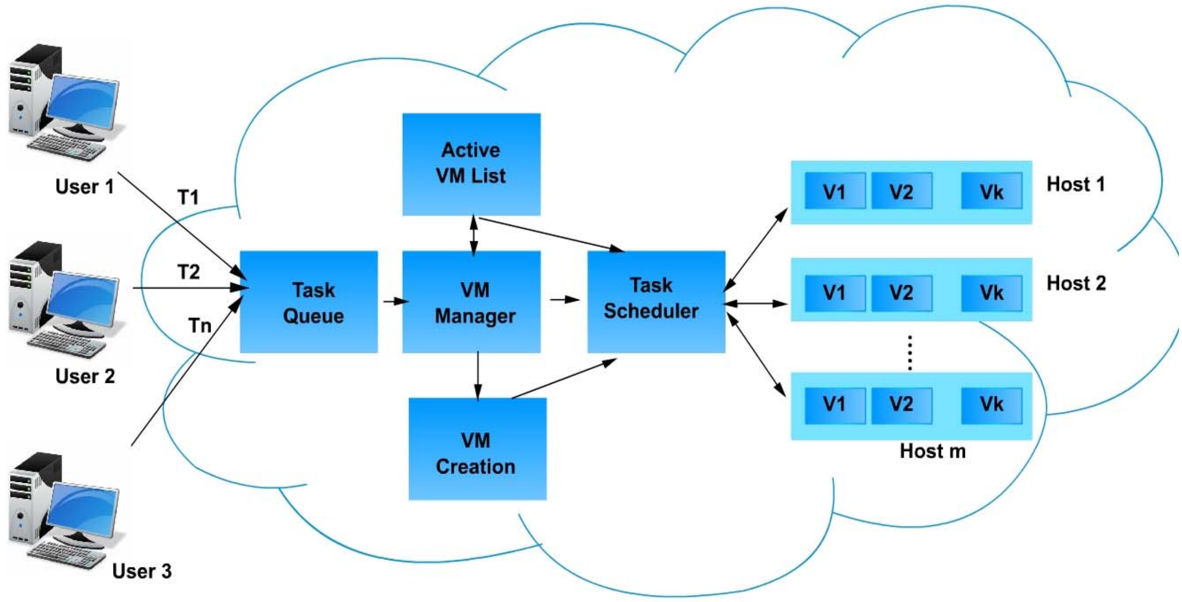


Fig 2: Load balancing

This study proposes and implements an Adaptive Neuro-Fuzzy Inference System (ANFIS)-QoS Aware Genetic Algorithm (GA) (ANFIS-GA) technique for fast selection of the best VM, which is described in the next section.

4. Proposed Methodology

The deployment of user applications in the cloud necessitates a variety of resources at various levels. As a result, the customer must buy the resources needed to meet the application's requirements.

4.1. *ANFIS Model*

For the QoS aware Genetic algorithm, the ANFIS rule [15] is used for parameter optimization. The adaptive neuro-fuzzy inference system is a piece of equipment that integrates fuzzy logic and artificial neural networks to make influential processing. For each of the input with 1 as the maximum value and 0 as the minimum value, two rules have been established in this article. It's a multilayer feed forward network where a specific function is performed on each node of the input signals.

The circle and square node symbols are used to denote different learning parameters. The parameters that must be changed to attain the required input–output characteristics are determined by the learning rules. The ANFIS architecture and inputs are shown in Fig 3. The ANFIS model takes as input parameters such as cost, energy, resource utilization, and QoS requirements. To achieve the best outcomes, these output parameters are fine-tuned utilizing the genetic algorithm (GA). The GA algorithm aids ANFIS in adjusting the membership functions.

It is stated that the fuzzy inference system contains five layers of adaptive network with 2 inputs a & b and only 1 output c. The function of the node in the j^{th} position of the l^{th} layer is indicated as $O_{l,j}$ are of the same function family as those of the node in a similar layer, as shown beneath:

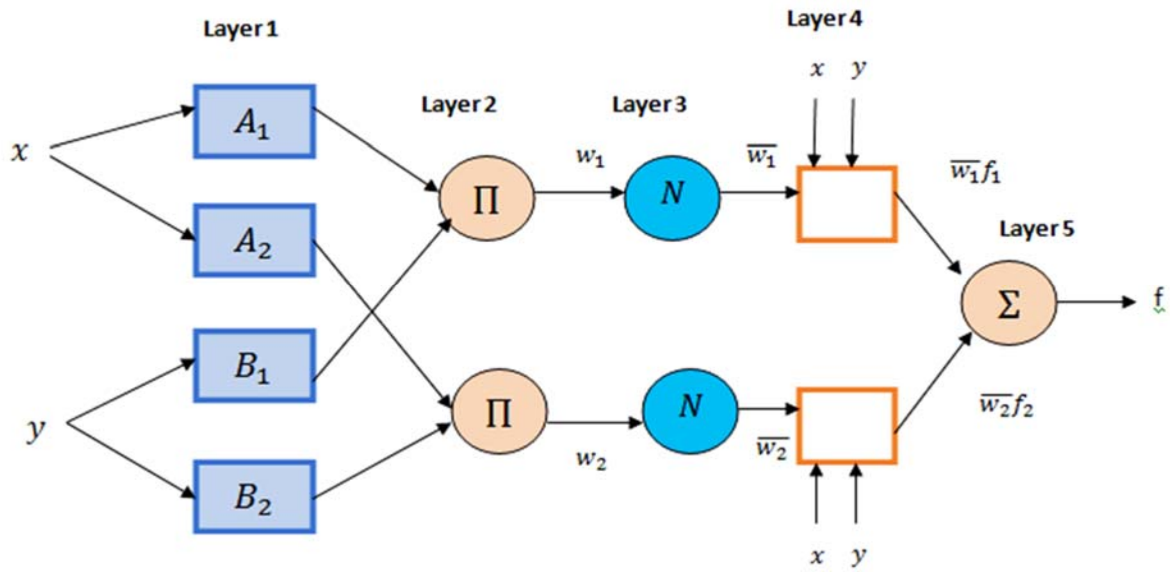


Fig 3: ANFIS Model

Layer 1: Input layer is denoted as layer 1. Every node j in layer 1 denotes the square node by the node function $O_{l,j}$ and the function of membership of A_j which defines the degree to which the presented influences the quantifier A_j. The input of the function of membership is bell-shaped membership function, with the highest value being 1 and 0 being the minimum value.

$$O_{1,j} = \mu A_j(x) \quad forj = 1,2$$

$$\mu A_j(x) = \frac{1}{1+\left[\left(x-\frac{r_j}{p_j}\right)^2\right]^{q_j}} \tag{1}$$

where r_j,p_j,q_j denotes the parameters, q denotes the (+) ve value & r indicates the centroid of the curve.

Layer 2: All nodes in this layer denotes the square node, denoted as Π, which generates input signals and sends the product output.

$$O_{2,j} = w_j = \mu A_j(x) = \mu B_j(y) forj = 1,2 \tag{2}$$

Layer 3: The square node, denoted by M, is shown by each node in this layer. According to the equation below, the jth node computes the ratio of the j^{th} rule's firing strength to the total of all the rules' firing strengths. This layer's output is stated as normalized firing strengths.

$$O_{3,j} = \underline{w_j} = \frac{w_j}{w_1+w_2} forj = 1,2 \tag{3}$$

Layer 4: The node function represents the square node in every node j in this layer. This layer's attributes are represented by succeeding attributes.

$$O_{4,j} = \underline{w_j} f_j = \underline{w_j}(t_j + u_j + v_j) \tag{4}$$

Where $t_j, u_j \& v_j$ denotes the features.

Layer 5: The only node in layer 5 denotes the circle node, denoted asΣ, that calculates the total output by adding all of the input signals as shown in the equation given.

$$O_{5,j} = \sum_j \underline{w_j} f_j = \frac{\sum_{j=1} w_j f}{\sum_{j=1} w_j} = \text{output} \quad (5)$$

4.2. *Requirements for QoS performance*

Several things must be considered while deploying the customer-satisfied application to the cloud. The database instance's reaction time and the instance of computing's response time are the two aspects in question. The database's response time is mostly determined by the anticipated demand (E_{dbd}). The passion distribution technique is used to model the database demand. Equation 7 depicts the response time of an application's database instance with a service rate ofS_{Rdb}.

$$T_{dbI}(S_{Rdb}) = \frac{(S_{Rdb})^{-1}}{1-\frac{E_{dbd}}{S_{Rdb}}} \quad (6)$$

The database instance's response time should not be shorter than T_{dbI}. T_{dbI} is the highest value stated by the client for database instance response time.

$$T_{dbI}(S_{Rdb}) \leq Rt_{db} \quad (7)$$

Another key concern when deploying an application to the cloud is the computer instance's reaction time. Assume that $MI_{Rt}$is the computing instances' minimum response time and E_{cd} is the predicted rate of arrival of the request for application to the computing instance. Equation depicts the model for computing instance response time.

$$T_c(MI_{Rt}) = \frac{(MI_{Rt})^{-1}}{1-\frac{E_{cd}}{MI_{Rt}}} \quad (8)$$

The compute instance's reaction time should not be shorter than Tc. The Tc is the customer-specified highest response time.

$$T_c(MI_{Rt}) \leq Rt_c \quad (9)$$

4.3. *Resource Allocation using QoS Aware Genetic Algorithm*

The algorithm's major goal is to improve the quality-of-service performance. The algorithm's result is ensured to be the prime optimal solution. The best solutions, on the other hand, are never-ending. Cloud service companies typically supply a limited number of computing and database instances. To locate appropriate cloud resources depending on the needs of the customer's application, algorithm 1 is designed. The Genetic algorithm was used to determine the best answers in this case. The genetic algorithm is regarded as one of the top effective methods for resolving problems of optimization such as resource allocation.

4.3.1. Chromosome Representation

We examined the customer QoS needs as well as the available Resources when creating the chromosomal representation. The chromosome representation in the genetic algorithm is shown in Equation.

$$\lambda(p,q) = Q_p(R_q),\ where p = 1, \dots n, q = 1, \dots m \quad (10)$$

where the chromosomal representation is $\lambda(p,q)$,the QoS needs are p and the available resources are q.

4.3.2. Initial Population

Population initialization is a significant aspect in the evolutionary algorithm, and an appropriate population results in a fine-tuned output. The population initialization in the genetic algorithm is represented in Algorithm 1.

4.3.3. Fitness Function

For each chromosome that has been initialized, the fitness function is determined. This fitness feature improves the solution's quality. The goal of the resource scheduling method is to make the resources respond as quickly as possible.

$$f = w_1(max(T_{dbI}(S_{Rdb}), T_c(MI_{Rt})) \quad (11)$$

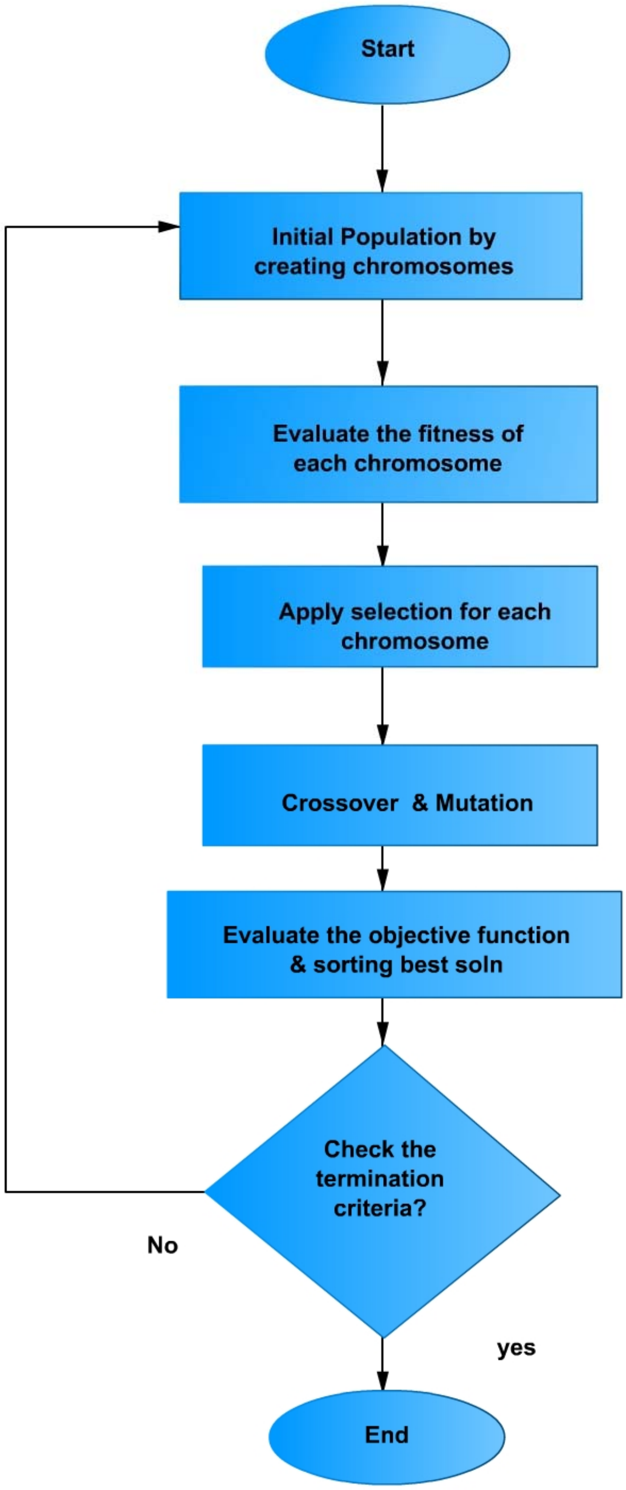


Fig 4: Flowchart for QoS Aware Genetic algorithm

Algorithm 1: ANFIS-GA algorithm

Input: QoS requirements, no. of iterations, deployment cost δ **, threshold, Resources A**
Output: Maximized QoS Performance
Step 1: Initialize Population
Begin
    for p in 1 to n do
        for q in 1 to n do
            $if(\delta > T)then$
                A=A-A$_q$
            else
                Compute Eqn 7 &Eqn 9
            End if
        End for
    End for
Step 2: while t<maximum no. of iterations
        for each chromosome
            Compute the fitness value using Eqn
            ANFIS rule is used to achieve best results
            if ($T_{dbI}(S_{Rdb}) \leq Rt_{db}$)**, then f is high**
            if ($T_{dbI}(S_{Rdb}) \geq Rt_{db}$)**, then f is low**
            if ($T_c(MI_{Rt}) \leq Rt_c$)**,then f is high**
            if ($T_c(MI_{Rt}) \geq Rt_c$)**,then f is high**
            End if
            End if
            End if
            End if
Step 3: **perform the crossover & mutation operation**
        End for
    record the optimal soln for current iteration
    t=t+1;
    if current iteration result is not same as last
    then k=0;
    else
    k=k+1;
    end if
end while

4.3.4. Cross over and mutation

Originally, 2 chromosomes are chosen at random from the Gϵ {0, 1} value assigned to each chromosome. For the chromosomes that are chosen with the probability Pc random number Gϵ {0,1} allocated to every chromosome, a cross over mechanism is used. If G < Pc, the chromosomes will crossover at one point. For the best results, the mutation level is kept low. The mutation is performed by changing the position of one single bit in the chromosomes.

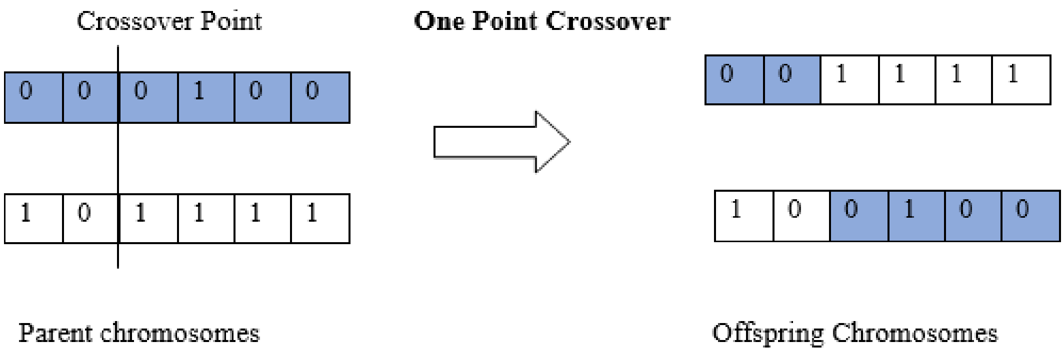


Fig 5: One Point Crossover

5. Results and Discussion

With regard to the evaluation metrics, this section elaborates on the results and discussion obtained utilizing the suggested ANFIS-GA. The experiment is carried out on a computing platform comprised of five machines each with 20 GB of RAM and 5 TB of storage. CloudSim is used to create the simulation environment. CloudSim's virtualization engine constructs the data centers. This section discusses the experimental findings.

5.1. Evaluation Metrics

The proposed optimization algorithm's performance is measured using performance metrics like utilization of memory, utilization of resources, skewness and utilization of CPU.

Utilization of resources: It is defined as the ratio of the number of allotted resources to the total number of resources available, denoted as

$$\alpha = \frac{A}{T} \tag{12}$$

where A is the number of resources allocated and T is the total number of resources.

Utilization of memory: It is defined as the ratio of the amount of memory required to complete a task to the total amount of memory existing in the cloud, which is computed as,

$$M = \sum_{i=1}^{y} \left(\frac{m_i}{a_i}\right) \tag{13}$$

where m_i is the memory required to accomplish the task and a_i is the total available memory.

Utilization of CPU: It is defined as the ratio of the number of CPUs requested to complete the work to the total number of CPUs available.

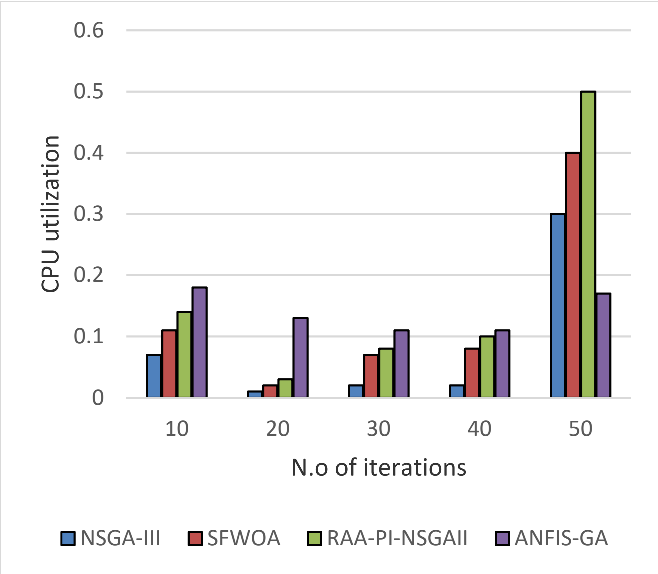
$$C = \sum_{i=1}^{y} \left(\frac{R_i}{N_i}\right) \tag{14}$$

where R_i is the CPU needed to accomplish the task and N_i is the total number of available CPU.

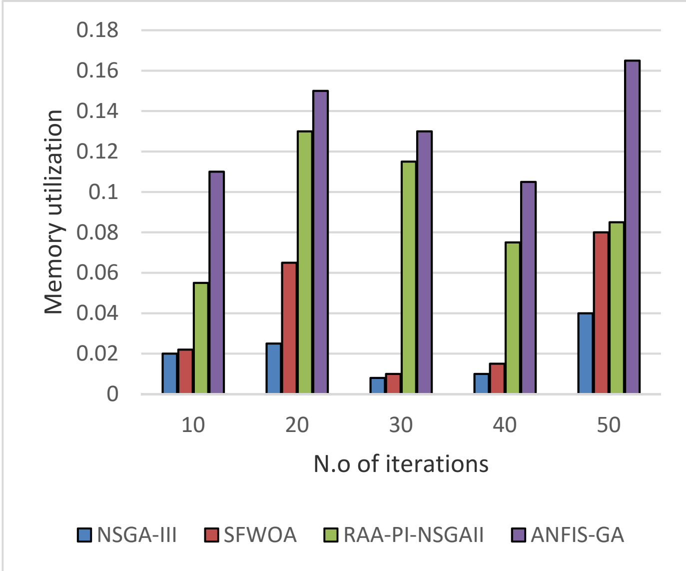
Skewness: It's a metric for detecting unequal resource use on a cloud server.

$$S_n = \left(\frac{A_n}{A} - 1\right) \tag{15}$$

where S_n is skewness and A_n is the average resource utilization.

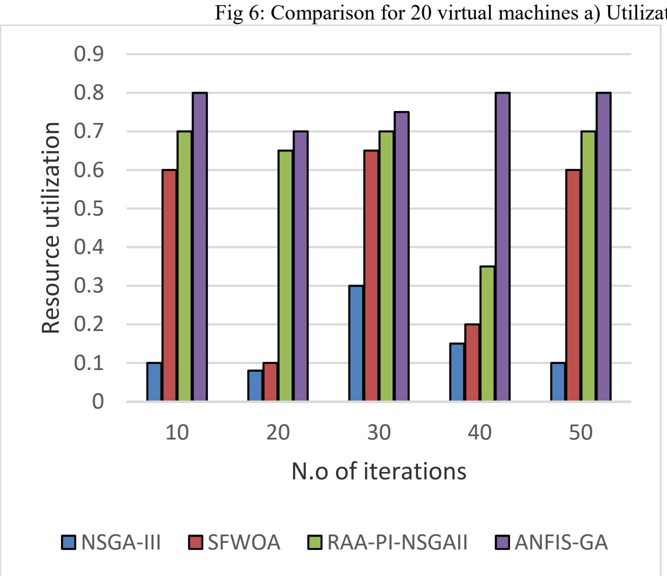


(a)

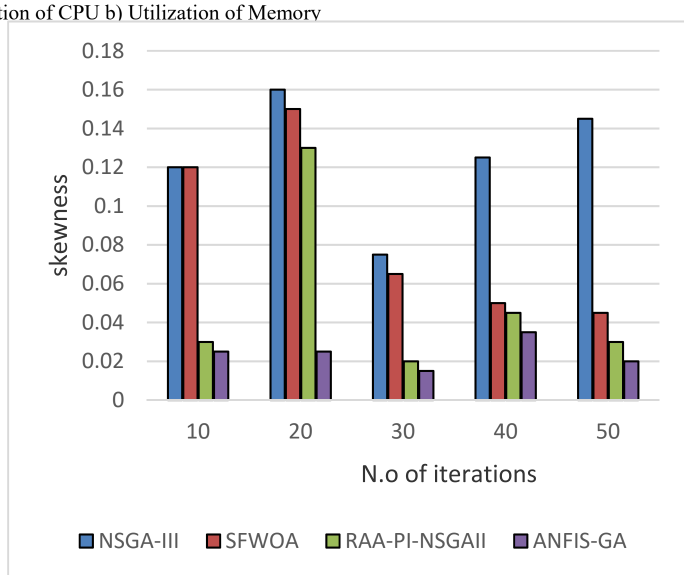


(b)

Fig 6: Comparison for 20 virtual machines a) Utilization of CPU b) Utilization of Memory



(c)



(d)

Fig 6: Comparison for 20 virtual machines a) Utilization of resources b) skewness**.**

Fig. 6 presents the results of a comparison analysis using the proposed method, which took into account 20 virtual machines. Fig. 6a shows the CPU consumption as a function of the number of iterations. When the number of iterations is set to 20, the CPU used by known algorithms such as NSGA-III, SFWOA, and RAA-PI-NSGAII is 0.0018, 0.021, and 0.0325, respectively, but the proposed ANFIS-GA used 0.1501.The utilization of memory use in relation to the number of iterations is shown in Fig 6b. When the number of iterations is set to 30, the memory used by current algorithms such as NSGA-III, SFWOA, and RAA-PI-NSGAII is 0.008, 0.010, and 0.115, respectively, but the proposed ANFIS-GA used 0.1308.

The utilization of resource use in relation to the number of iterations is shown in Fig 6c. When the number of iterations is set to 20, the resources used by known techniques such as NSGA-III, SFWOA, and RAA-PI-NSGAII are 0.0087, 0.010, and 0.6594, respectively, but the proposed ANFIS-GA used 0.7206.The study of skewness in relation to the number of iterations is shown in Fig 6d. When the number of iterations is set to 40, the skewness produced by current techniques such as NSGA-III, SFWOA, and RAA-PI-NSGAII is 0.1259, 0.0508, and 0.0302, respectively, whereas the suggested ANFIS-GA obtained 0.0209.

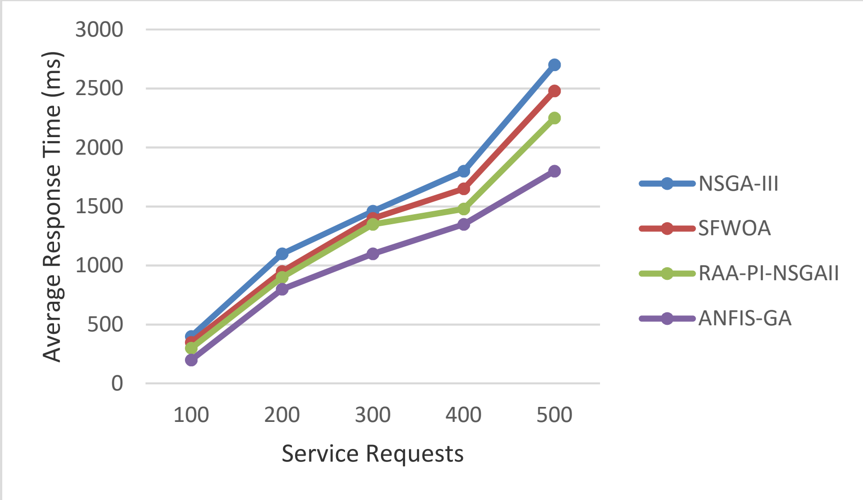


Fig 7: Average Response Time

The proposed work's average service response time is examined and compared to existing approaches. The proposed work is demonstrated to have a minimal service response time based on the experimental results. The optimal resource selection, which significantly reduces time consumption, is the fundamental cause for the short service response time.

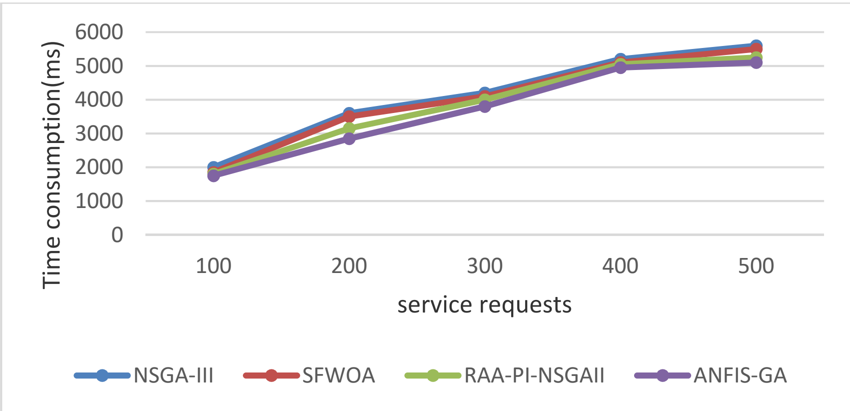


Fig 8: Analysis of Execution time

The proposed work's average service response time is examined and compared to existing approaches. The proposed work is demonstrated to have a minimal service response time based on the experimental results. The optimal resource selection, which significantly reduces time consumption, is the fundamental cause for the short service response time.

6. Conclusion

To achieve the ideal settings in the cloud server, we created the fuzzy-based QoS Aware GA optimization method in this study. The method creates a suitable server for the virtual machine based on the rule scheme for improved CPU and memory usage. The key problem in the cloud system is allocating resources while meeting client QoS requirements while staying within limited cost. Customers can select the best resources for their application deployment by contacting cloud service providers. For determining the optimal discrete solution to the issue, the suggested technique employs the genetic algorithm. The experiments were carried out on a variety of workloads utilizing a real-time cloud service provider. The findings showed that the suggested technique is effective at maximizing the QoS performance trade-off with a less execution time of 4%, 8%, 11% when compared with RAA-PI-NSGAII, SFWOA, NSGA-III with less computational cost.

Conflict of interest

"The authors have no conflict of interest to declare"

References

[1] Afrin, M., Jin, J., Rahman, A., Rahman, A., Wan, J. and Hossain, E., 2021. Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey. *IEEE Communications Surveys & Tutorials*.

[2] Bouaziz, A., Saddoud, A. and Chaouchi, H., 2021. QoS-aware resource allocation and femtocell selection for 5G heterogeneous networks.

[3] Chen, J., Du, T. and Xiao, G., 2021. A multi-objective optimization for resource allocation of emergent demands in cloud computing. *Journal of Cloud Computing*, *10*(1), pp.1-17.

[4] Gong, S., Yin, B., Zheng, Z., & Cai, K.-y. (2019). Adaptive multivariable control for multiple resource allocation of service-based systems in cloud computing. *IEEE Access, 7,* 13817–13831.

[5] Li-Der Chou, L. D., Chen, H.-F., Tseng, F.-H., Chao, H.-C., & Chang, Y. J. (2018). DPRA: Dynamic power-saving resource allocation for cloud data center using particle swarm optimization. *IEEE Systems Journal, 12*(2), 1554–1565.

[6] Liu, X., Zhang, X., Li, W., & Zhang, X. (2017). Swarm optimization algorithms applied to multiresource fair allocation in heterogeneous cloud computing systems. *Computing, 99*(12), 1231–1255.

[7] Miriam, A.J., Saminathan, R. and Chakaravarthi, S., 2021. Non-dominated Sorting Genetic Algorithm (NSGA-III) for effective resource allocation in cloud. *Evolutionary Intelligence*, *14*(2), pp.759-765.

[8] Pradhan, A., Bisoy, S.K. and Das, A., 2021. A survey on PSO based meta-heuristic scheduling mechanism in cloud computing environment. *Journal of King Saud University-Computer and Information Sciences*.

[9] Sharma, N. and Kumar, K., 2021. Resource allocation trends for ultra dense networks in 5G and beyond networks: A classification and comprehensive survey. *Physical Communication*, *48*, p.101415.

[10] Subhash, L.S. and Udayakumar, R., 2021. Sunflower whale optimization algorithm for resource allocation strategy in cloud computing platform. *Wireless Personal Communications*, *116*(4), pp.3061-3080.

[11] Wang, W., Jiang, Y., & Weiwei, W. (2017). Multiagent-based resource allocation for energy minimization in cloud computing systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems, 47*(2), 205–220.

[12] Wei, W., Yang, R., Gu, H., Zhao, W., Chen, C. and Wan, S., 2021. Multi-objective optimization for resource allocation in vehicular cloud computing networks. *IEEE Transactions on Intelligent Transportation Systems*.

[13] Xu, Y., Gui, G., Gacanin, H. and Adachi, F., 2021. A survey on resource allocation for 5G heterogeneous networks: Current research, future trends and challenges. *IEEE Communications Surveys & Tutorials*.

[14] Yahia, H.S., Zeebaree, S.R., Sadeeq, M.A., Salim, N.O., Kak, S.F., Adel, A.Z., Salih, A.A. and Hussein, H.A., 2021. Comprehensive survey for cloud computing based nature-inspired algorithms optimization scheduling. *Asian Journal of Research in Computer Science*, pp.1-16.

[15] YR, S.K. and Champa, H.N., 2021. An Energy Aware Data Scheduling Approach in Cloud Using GK-ANFIS. *International Journal of Computer Networks and Applications*, *8*(5), pp.490-506.

Authors Profile

D. Komalavalli is a Research Scholar at the Bharathiar University, Coimbatore, India. She received her Postgraduate in Computer Applications from the Indira Gandhi National Open University, India and Computer Science and Engineering from the Anna University, Coimbatore. Her research interests include cloud computing, virtualization and swarm intelligence. She has nine years of teaching experience. She is a life member of Indian Society for Technical Education.

T. Padma is a Professor and Head at the Department of Computer Applications, Sona College of Technology, India. Her research interests include machine learning, data analytics and interaction design. She has executed research projects as Principal Investigator funded by AICTE, UGC and NCW; published around 60 journal papers, books and book chapters. She is very instrumental in setting up the institution's management information system; Certified Six Sigma Black Belt; recipient of Shayesta Akhtar Memorial National Award of the ISTE in 2015. Her biography was included in the 2010 edition of the Marquis Who's Who in the World, New Jersey, USA.