International Journal of Innovative Technology and Exploring Engineering (IJITEE)
ISSN: 2278-3075 (Online), Volume-13 Issue-11, October 2024

Enhanced Ant Colony Based VM Selection and Consolidation for Energy Conservation

Usha Kirana S P

Abstract: ***Cloud Computing (CC) involves extensive data centers with numerous computing nodes that consume significant electrical energy. Researchers have identified high service-level agreement (SLA) violations and excessive energy consumption (EC) as major challenges in CC. Traditional approaches often focus on reducing EC but tend to overlook SLA violations, particularly when selecting Virtual Machines (VMs) from overloaded hosts. To address these issues, this paper introduces the Enhanced Ant Colony Optimization (EACO) algorithm, aims to reduce high EC and SLA violations by utilizing a unique approach where the best-performing ant explores movement patterns and identifies distances between movements. The algorithm comprises three key steps: tracking pheromone trails, updating pheromones and selecting the cities (VMs). The effectiveness of EACO was validated through simulations using CloudSim. Compared to existing techniques, EACO demonstrated a significant reduction in EC, achieving approximately 41-44% lower energy consumption than the traditional Ant Colony Optimization (ACO) algorithm when applied to Planet Lab data. This suggests that EACO offers a more efficient and stable solution for managing EC and SLA violations in cloud environments.***

Keywords: ***VM Consolidation, Energy Conservation and Enhanced ACO.***

I. INTRODUCTION

Cloud computing (CC) leverages the internet to connect vast amounts of storage and computing resources [1]. Users pay based on their actual resource usage due to the on-demand nature of CC services [2]. To meet the growing demand for resources, large-scale data centers are essential [3]. However, these data centers require substantial electrical power, which increases both computational and operational costs [4]. The Energy Efficiency and Performance Trade-offs are based on Consolidation mechanism. VM (Virtual Machine) consolidation is a strategy used to convert energy efficiently in cloud data centers [5]. Although effective in reducing energy consumption, aggressive VM consolidation can degrade performance. This is due to the shared nature of physical resources among VMs, which may lead to increased response times, especially when applications encounter unexpected resource demands [6].

Manuscript received on 26 August 2024 | Revised Manuscript received on 02 September 2024 | Manuscript Accepted on 15 October 2024 | Manuscript published on 30 October 2024.
**Correspondence Author(s)*
Usha Kirana S P*, Department of Computer Science Engineering, DBIT, Bangalore (Karnataka), India. E-mail: usha14.nayak@gmail.com, ORCID ID: 0000-0002-3881-6115

Cloud Services and Revenue Generation are the concerns of the cloud providers. Even though cloud degradation, SLA violations persist under varying workloads, necessitating a balance between performance and energy consumption to minimize EC and meet SLA obligations [7][14][15][16][17][18]. This research introduces an enhanced version of the Ant Colony Optimization (ACO) algorithm, termed Enhanced ACO (EACO), aimed at reducing SLA violations and minimizing high energy consumption. The ACO algorithm simulates ant behaviour, optimizing the path from the nest to food sources based on pheromone trails [13]. The proposed EACO method extends traditional ACO by incorporating an additional term, 'depth,' which enhances the recursion process. Providers offer three major types of services: Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) [8]. Users must sign Service Level Agreements (SLAs) with cloud providers before accessing these services [9]. Therefore, it is crucial for providers to deliver high Quality of Service (QoS) to meet SLA requirements [10].

The Methodology and Evaluation involves the VM allocations which are optimized using the Inter Quartile Range (IQR) technique. Simulations assess the EACO's effectiveness in reducing energy consumption, VM migrations and the number of host shutdowns. The paper organized as 1. Section 2: Reviews existing techniques and their limitations related to SLA violations in cloud computing. 2. Section 3: Describes the system model and problem formulation. 3. Section 4: Explains the proposed EACO methodology. 4. Section 5: Compares the performance of EACO with traditional approach ACO. 5. Section 6: Concludes the research and outlines the future work. This section reviews various techniques developed to address Service Level Agreement (SLA) violations and high energy consumption (EC) in cloud computing environments. Each method's benefits and limitations are discussed to provide context for the proposed Enhanced-Ant Colony Optimization (EACO) algorithm.

II. LITERATURE SURVEY

This section reviews various techniques developed to address Service Level Agreement (SLA) violations and high energy consumption (EC) in cloud computing environments. Each method's benefits and limitations are discussed to provide context for the proposed Enhanced-Ant Colony Optimization (EACO) algorithm.

The MeMs method uses M-estimate regression to calculate an upper threshold for CPU usage on hosts.

If a host's CPU usage exceeds this threshold, it is considered overloaded. The MuMs VM selection strategy then determines which VMs to migrate to improve overall consolidation and balance the load across hosts [11]. Linear Regression Migration Time applies simple regression models to subsets of CPU usage data to estimate a curve that represents actual CPU usage. This model incorporates the time required for VM migration, aiming to optimize the trade-off between effective resource utilization and minimal migration impact [11]. Modified Adaptive Migration adjusts the threshold for high CPU usage based on the variance in CPU usage among VMs. It selects the VM with the lowest CPU usage for migration, aiming to reduce overall host utilization. Repeated migrations are performed until the host utilization falls below the designated limit [12]. IQR (Interquartile Range) is a statistical measure of dispersion, calculated as the difference between the third quartile (Q3) and the first quartile (Q1) of data. It is preferred over the full range due to its robustness against outliers. The IqrMc method uses IQR to identify and migrate VMs that have a strong correlation in CPU usage with other VMs, helping to maintain balanced performance [12].

III. SYSTEM MODEL

The proposed work designs an EACO with the goal of maximizing resource utilization and reducing SLA breaches. In Figure 1, the EACO operating method is shown for the energy conservation. The Ants' social behavior is utilised in the suggested EACO approach. The EACO, which is a modified version of the ACO algorithm, uses the pheromones to come up with unique answers for repetitive process.

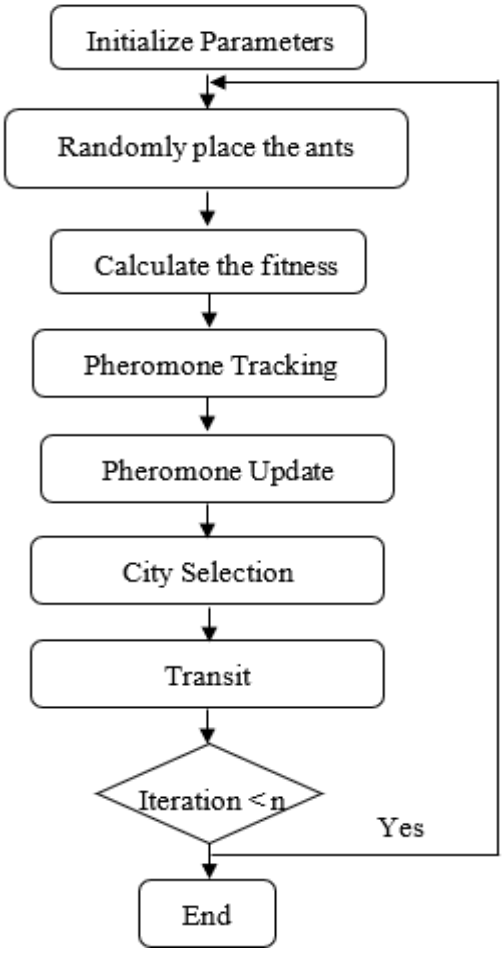

Figure 1: Working Principle of Proposed Method

Following the random placement of each ant, the fitness function is calculated. The fundamental ACO model reveals that no solution with an extremely high pheromone trail intensity can be obtained if the number of iterations is decreased. In order to solve the VM consolidation problem, the EACO algorithm was created.

A. Pheromone Tracking

Pheromone tracking in EACO involves the following aspects:

- Pheromone Intensity: Ants uses the intensity of pheromones to decide their movement between cities from source to destination (or, in the case of VM placement, between VMs and PMs). Higher pheromone intensity generally indicates a more favorable or promising path, guiding ants towards those options.
- Potential for Unlimited Values: In practice, the number of pheromone values for each path (city) is potentially very large, given the numerous possible VM and PM assignments. However, due to the probabilistic nature of ant movements, the system focuses on paths with significant pheromone levels.

B. Pheromone Updating

The process of updating pheromones in the EACO algorithm can be detailed as follows:

- Limited Number of Ants: Since the number of ants is often much smaller than the number of possible paths or cities, updating pheromones for every possible path is impractical.
- Selective Updating: Instead of updating pheromones for all paths, the algorithm updates pheromone levels only for the paths that ants have actually travelled. This is done to reflect the quality of solutions found and to encourage paths that led to better solutions.

Updating Strategy:

- Increased Pheromone: On paths that resulted in high-quality pheromone amount leading to good resource utilization or minimal SLA violations.
- Evaporation: Gradually reduce the pheromone levels on paths that are less frequently used or did not contribute to good solutions, thus allowing exploration of new paths.

C. City Selection After Each Depth

In the context of optimization algorithms of ACO and EACO:

- Local Maxima: Cities (or solutions) might appear to be optimal in a local sense but are not necessarily the best overall. Ants might converge on these local maxima, potentially missing out on better global solutions.
- Depth-Based Exploration: EACO uses an iterative approach where ants explore deeper into the solution space over successive iterations. By doing this, the algorithm can overcome local maxima:
- Initial Depth: Ants might initially focus on local maxima.

Retrieval Number: 100.1/ijitee.K997513111024
DOI: 10.35940/ijitee.K9975.13111024
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

- Increased Depth: As the algorithm progresses, it explores more extensively, allowing for the identification and escape from local maxima. This helps in finding better global solutions.

Strategies are used to Avoid Local Maxima are

- Diverse Ant Paths: Encourage ants to explore a variety of paths to avoid getting stuck in local maxima.
- Pheromone Recalibration: Adjust pheromone levels to balance between exploiting known good paths and exploring new ones.

Let ants be *m* and *n* cities; the cities are fully connected to the edges *En*. The following are the steps in optimization:

1. **Initialization:** The pheromone level is shown as a minor positive variable and the m ants are arranged at random in the n cities.
2. **Path Construction:** In the ant colony system, the city is selected using what is known as the pseudo-random proportional rule, which governs state transition. According to this rule, an ant *k* chooses tuples s in the manner shown in the expression in equation (1), which discovers its path:

$$S = \begin{cases} argmax\ u \in T_K\{[\tau_u] \cdot [\eta_u]^\beta\}, & q \le q_0 \\ S & , \ q > q_0 \end{cases} \quad \text{.....(1)}$$

where β is the factor that affects the number of pheromones, τ is the number of pheromones and η is the heuristic values of certain tuples as described in equation (1). The untraversed tuple that an ant k attempts to go through in tuple T is denoted as Tk ∈ T. In the tuple, q ∈ [0, 1]. The random variable T has a uniform distribution, while the random variable S is selected using the probability distribution formula (2). Ants balance their exploitation and exploration activities by using the state transition rule to decide which city or node to visit next.

Each ant chooses which city to visit next based on the transition probability, which is given by equation (2)

$$p(i,j) = \begin{cases} \frac{[\tau(i,j)]^\alpha [\eta(i,j)]^\beta}{\sum_{u \in J}[\tau(i,u)]^\alpha [\eta(i,u)]^\beta}, & if\ j \in J \\ 0, & otherwise \end{cases} \quad \text{.....(2)}$$

The heuristic edge value is η(i, j), the two weighting factors that control the relative significance of pheromone and heuristic information are α and β, the set of cites J is not visited, and ant k is in city i and the next city is j.

Ants are more likely to select edges that other ants have already selected when the value of is high, indicating that the trail is highly significant.

3. **Proceed to Step** 4 if every ant has the full traversal path to every city. If not, proceed to step 2.
4. **Local Updating Rule:** Throughout a single cycle, different ants move from one node to another, continuously updating the pheromone trail between them. This pheromone trail update is referred to in the local updating rule. Equation (3) is used to update pheromone in accordance with the rule, which is represented by the formula provided in equation (3).

$$\tau(i,j) \leftarrow (1-\rho).\tau(i,j) + \rho.\Delta\tau(i,j) \quad \text{.....(3)}$$

where the value of Δτ (i,j) is related to the fitness function value and ρ ∈ (0,1) represents the evaporation rate.

Global update rule: The ant with the shortest tour from the start of the trial is the only one allowed to deposit pheromone in ACS, making them the best ants at global level. This update is performed after each ant has finished its tour.

5. Establish the stopping criteria as the difference, as given by equation (4), between the weak ant's path Y= (y_1, $y_{2....}$,y_s) and the best ant's route X=(x_1,x_2,…,x_r)

$$d = \sum_{i=1}^{n} |X_i - Y_i| \quad \text{.....(4)}$$

Where, the values of the i^{th} variable at locations X_i *and* Y_i corresponds to the number of variables, n.

6. The process ends if the stopping condition is satisfied; if not, the ants are arranged at random and the best solution is found.

i. VM Selection Process

In the VM selection process, the aim is to map Virtual Machines (VMs) to Physical Machines (PMs) in a way that ensures balanced resource utilization across the PMs. This is part of an optimization problem often addressed in systems like load balancing and resource allocation.

ii. VM Placement

The initial allocation of Virtual Machines (VMs) to Physical Machines (PMs) is done randomly. This provides a starting point for the optimization process. Workloads on VMs can change over time, which affects the resource utilization on PMs. This necessitates an adaptive approach to maintain efficient resource usage.

The Local Agent (LA) continuously monitors the current resource utilization of each PM. Specifically, it tracks metrics such as CPU usage. Based on the observed resource utilization, the LA categorizes PMs into three states (1) Normal: The PM's resource usage is within acceptable limits. (2) Overloaded: The PM's resource usage exceeds its capacity. (3) Under-loaded: The PM has resources available beyond its normal operating requirements.

The LA provides this categorized data to the Global Agent (GA) for further processing. The Global Agent (GA) uses the EACO algorithm to develop a global migration plan based on the data received from LAs. This plan aims to optimize resource utilization by redistributing VMs among PMs. The GA instructs the Virtual Machine Manager (VMM) to execute the migration plan. This involves moving VMs to achieve a more balanced distribution of resources across PMs. The Virtual Machine Manager (VMM) performs the actual migration of VMs according to the consolidation plan developed by the GA. This step is required for implementing the optimized resource allocation. The EACO algorithm helps to effectively manage the VM-to-PM allocation by continuously adapting to changing workloads. This leads to significant improvements in: the system reduces unnecessary energy usage and Improved resource distribution minimizes the likelihood of SLA breaches by ensuring that PMs do not become overloaded. In the subsequent chapter, the effectiveness of the proposed EACO method is validated. This likely involves testing and comparing the performance of the EACO-based system against other approaches to confirm its benefits in real-world scenarios.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

iii. Construction of EACO Algorithm

Algorithm 1 shows the pseudocode of an EACO algorithm. Ant pheromone deposition on fluid constituents is simulated using an Nv * Np pheromone matrix. Every cycle begins with an empty solution, a set of PMs and a set of VMs that have been rearranged at random for each ant. For every ant, the VM set is shuffled to randomize the search in the solution space. To build their solutions, each ant follows the ACS rule. For each iteration, each ant is chosen at random and is permitted to choose a virtual machine (VM) from among all the VMs to place next to its existing PM. If the current PM is completely used up or there isn't a viable virtual machine left, another PM is selected to take over. When solving the problem, the selected ant uses the pseudo-random-proportional rule. This rule is based on heuristic information $\eta(i,j)$ and the current pheromone concentration $\tau(i,j)$ on the (VM, PM) pair. It instructs the ant to choose the virtual machines (VMs) from the available resources that will ultimately lead to a better usage of PM resources. Because of this, the ant is more likely to choose the heuristic values and (VM, PM) pair with higher pheromone concentrations. An ant gets eliminated from its virtual machine list (antRecord) after it has finished adding all of the virtual machines (VMs) to its temporary list of active ants.

When all the ants have completed building their solutions (a cycle is complete), all the solutions computed by the ants in the current cycle are compared to the global-best-solution (GBS) that has been identified so far and their obtained values. When the quality of the answer does not improve in the subsequent iterationEnd cycles, the algorithm terminates. The different elements of the EACO algorithm are formally defined in the remainder of this section.

Algorithm 1: EACO Algorithm

```
Input: Set of PMs, Set of VMs, Set of ants antGroup
 Output: Global-best-solution (GBS)
Initialization:
        Initialize pheromone values for each (VM, PM)
pair
        Initialize pheromone matrix
        Initialize antGroup
        Set optimum solution to NULL
For each iteration until stopping criterion is met:
   For each ant in antGroup:
        Select a VM to PM assignment based on
pheromone trails
        Compute fitness of the solution
        If current solution is better than the optimum:
            Update optimum with current solution
        Update the ant's personal best
        Update the global best with the best of all ants
Return GBS
```

iv. VM Migration and Consolidation Process

Local search is employed to adjust and refine an infeasible solution, making it feasible and improving its quality before any global pheromone updates occur. This search is performed on the current solution when an infeasible state is detected, i.e., when resource constraints are violated or PMs are overloaded. The local search consists of balancing the load on PMs that is migration operation reduces the load on an overloaded PM by redistributing VMs to other PMs. VMs are selected from the overloaded PM and moved to other PMs with lower utilization. This helps in balancing the resource usage across the system. The selection of VMs for migration and the destination PMs are typically guided by the current resource utilization metrics and optimization goals.

IV. MIGRATION OPERATION

The primary goal is to balance the load on servers by moving VMs from servers that are overloaded to those that are not overloaded but have sufficient resources (RAM and CPU) to accommodate the incoming VMs. Each VM on an overloaded server is assessed to determine if it can be migrated. For each VM, the process checks whether the intended unloaded server has enough available RAM and CPU resources to support the VM. Each VM is compared against every VM on the unloaded server. This involves evaluating and ensuring the intended server has sufficient resources. Considering the time required for migration Minimum Migration Time (MMT) to ensure its feasibility and efficiency. Once a feasible target server is identified for a VM, the migration is planned based on the resource availability and MMT. The migration of VMs is carried out according to the plan. This helps to redistribute the load and reduce the strain on overloaded servers. The migration process is complete when: all previously overloaded servers is balanced and brought to a normal level and Constraints Met: The migration cannot proceed further if no more feasible VM movements are possible due to resource constraints or other limitations. The migration process is carried out after the Order Exchange Procedure. The Order Exchange Procedure likely involves rearranging or reassigning VMs in a way that prepares the system for more effective migration. This procedure helps to optimize the initial VM assignments, which can make subsequent migrations more efficient. While local search aims to refine and improve solutions, it may not always be able to transform an infeasible solution into a feasible one initially. As the local search progresses and gets closer to an optimal solution, previously infeasible solutions may become feasible. This potential for improvement is inherent in the iterative nature of local search methods.

A. Implementation and Experiments

The parameters, datasets, and metrics utilized to illustrate the effectiveness of the suggested mechanism are all described in the current section.

i. Experimental Setup

The proposed algorithm's performance is tested using the cloud computing simulation environment cloudsim. The HP ProLiant ML110 G5, a host instance type, features two processing cores, four gigabytes of RAM and 2660 MIPS of processing power. In the section below, the total number of VM migrations, host shutdowns, SLA violations and EC are used to compare the proposed EACO approach with the existing Ideal-ACO algorithm.

Retrieval Number: 100.1/ijitee.K997513111024
DOI: 10.35940/ijitee.K9975.13111024
Journal Website: www.ijitee.org

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

B. Analysis of Proposed EACO in PlanetLab Data

The proposed EACO performance is tested with the ideal ACO algorithm through the use of PlanetLab data in the experiments. Figure 2 displays the EC of proposed technique, while Table 1 displays the verified performance of the proposed strategy in terms of EC, number of host shutdowns, SLA violations and VM migrations.

Table 1: Performance Analysis of Proposed EACO for Planet Lab Data

Methods	Energy Conservation (kWh)	SLA Violation	No. of Host Shutdowns	No. of VM Migration
Ideal-ACO	85	1.5	1800	800
Proposed EACO	49	0.9	1600	600

i. Performance analysis of EACO with existing Ideal-ACO

Figure 2 shows the performance analysis of the EACO with the existing Ideal-ACO. Table 1 shows that, using the PlanetLab data, the suggested EACO's EC performs better (49kWh) than the ideal-ACO's energy consumption (85kWh) and the SLA violation is decreased from 1.5 to 0.9.

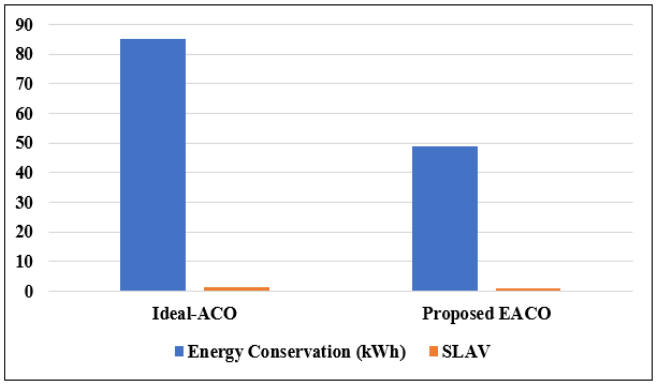

Figure 2: EACO Technique Performance Analysis of EC and SLA Violation

Figure 3 shows the Performance study of EACO with the current Ideal-ACO of host shutdown and number of VM migration. Table 1 shows that, in comparison to ideal-ACO, the proposed EACO method decreased the number of hosts shut down by 1600 and the number of virtual machine migrations by 600.

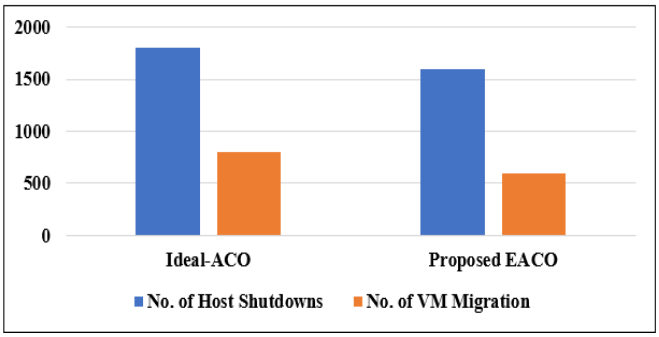

Figure 3: Evaluation of the EACO Approach's Performance with the Ideal-ACO

V. CONCLUSION

Optimization between energy consumption (EC) and Quality of Service (QoS) is a significant challenge for service providers and cloud users in data centers. This paper introduces the EACO (Enhanced Ant Colony Optimization) algorithm, which aims to strike an optimal balance between EC and QoS. Enhanced ACO offers several advantages over existing optimization techniques. It does not require initial values for task assignment to virtual machines (VMs) and performs a search within user-defined ranges. The performance of EACO was evaluated through simulations using CloudSim, comparing with the established ideal-ACO technique. The results demonstrate that EACO is more efficient than the ideal Ant Colony Optimization. Specifically, EACO consumed 49 kWh of energy for Planet Lab datasets while the ideal ACO consumed 85 kWh under the same conditions. However, EACO also resulted in a higher number of host shutdowns (1600) in the Planet Lab dataset due to the presence of various instance types with different Million Instructions Per Second (MIPS) values. Future work will focus on developing an enhanced optimization algorithm to address the issue of excessive host shutdowns observed with Planet Lab data, aiming to further improve efficiency and reduce energy consumption while maintaining high QoS.

DECLARATION STATEMENT

I must verify the accuracy of the following information as the article's author.

- **Conflicts of Interest/ Competing Interests:** Based on my understanding, this article has no conflicts of interest.
- **Funding Support:** This article has not been funded by any organizations or agencies. This independence ensures that the research is conducted with objectivity and without any external influence.
- **Ethical Approval and Consent to Participate:** The content of this article does not necessitate ethical approval or consent to participate with supporting documentation.
- **Data Access Statement and Material Availability:** The adequate resources of this article are publicly accessible.
- **Authors Contributions:** The authorship of this article is contributed solely.

REFERENCES

1. SS Gill, L Chana, M Singh and R Buyya. CHOPPER, "An intelligent QoS-aware autonomic resource management approach for cloud computing", *Cluster Comput.* 2018; 21, 1203-41. https://doi.org/10.1007/s10586-017-1040-z
2. Y Wang, X Tao, F Zhao, B Tian and AMVV Sai, "SLA-aware resource scheduling algorithm for cloud storage" *EURASIP J. Wireless Comm. Network.* 2020; 2020, 6. https://doi.org/10.1186/s13638-019-1604-0
3. J Prassanna and N Venkataraman, "Adaptive regressive holt-winters workload prediction and firefly optimized lottery scheduling for load balancing in cloud" *Wireless Network.* 2019; 27, 5597-615. https://doi.org/10.1007/s11276-019-02090-8
4. MH Malekloo, N Kara and ME Barachi," An energy efficient and SLA compliant approach for resource allocation and consolidation in cloud computing environments" *Sustainability Comput. Informat. Syst.* 2018; 17, 9-24. https://doi.org/10.1016/j.suscom.2018.02.001
5. M Ranjbari and JA Torkestani, "A learnining automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

data centers" *J. Parallel Distr. Comput*. 2018; 113, 55-62. https://doi.org/10.1016/j.jpdc.2017.10.009
6. CD Martino, S Sarkar, R Ganesan, ZT Kalbarczyk and RK Iyer,"Analysis and diagnosis of SLA violations in a production SAAS cloud", *IEEE Trans. Reliab*. 2017; 66, 54-75. https://doi.org/10.1109/TR.2016.2635033
7. SK Panda and PK Jana," SLA-based task scheduling algorithms for heterogeneous multi-cloud environment" *J. Supercomput*. 2017; 73, 2730-62. https://doi.org/10.1007/s11227-016-1952-z
8. M Kumar and SC Sharma," PSO-based novel resource scheduling technique to improve QoS parameters in cloud computing" *Neural Comput. Appl. 2019;* 32, 12103-26. https://doi.org/10.1007/s00521-019-04266-x
9. A Ramegowda, J Agarkhed and SR Patil, "Adaptive task scheduling method in multi-tenant cloud computing. *Int. J. Inform. Tech*" 2019; 12, 1093-102. https://doi.org/10.1007/s41870-019-00389-5
10. D Komarasamy and V Muthuswamy, "ScHeduling of jobs and adaptive resource provisioning (SHARP) approach in cloud computing" Cluster *Comput*. 2018; 21, 163-76. https://doi.org/10.1007/s10586-017-0976-3
11. X Zhou, K Li, C Liu and K Li, "An experience-based scheme for energy-SLA balance in cloud data centers" *IEEE Access* 2019; **7**, 23500-13. https://doi.org/10.1109/ACCESS.2019.2899101
12. L. Adhianto, ''Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers,'' Concurrency Comput. Pract. Exper., vol. 22, no. 6, pp. 685–701, 2010.
13. M. Dorigo and L.M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, 1997, Volume 1, Issue 1, Pages: 53–66. DOI: 10.1109/4235.585892. https://doi.org/10.1109/4235.585892
14. G. Sumathi, S.Rajesh, Optimization of Tasks using Scheduling Algorithms in Cloud Computin. (2019). In International Journal of Innovative Technology and Exploring Engineering (Vol. 8, Issue 6S4, pp. 1239–1245). https://doi.org/10.35940/ijitee.f1254.0486s419
15. Saravanan, A., & Murali, M. (2020). Exploration of Task Scheduling Algorithms in Cloud Computing Environments. In International Journal of Recent Technology and Engineering (IJRTE) (Vol. 8, Issue 5, pp. 4830–4833). https://doi.org/10.35940/ijrte.e6917.018520
16. Saravanan, N. P., & Kumaravel, T. (2019). An efficient Task Scheduling Algorithm using Modified Whale Optimization Algorithm in Cloud Computing. In International Journal of Engineering and Advanced Technology (Vol. 9, Issue 2, pp. 2533–2537). https://doi.org/10.35940/ijeat.b3813.129219
17. Jain, N., & Kumar, R. (2022). A Review on Machine Learning & It's Algorithms. In International Journal of Soft Computing and Engineering (Vol. 12, Issue 5, pp. 1–5). https://doi.org/10.35940/ijsce.e3583.1112522
18. Sharma, P. (2023). Advancements in OCR: A Deep Learning Algorithm for Enhanced Text Recognition. In International Journal of Inventive Engineering and Sciences (Vol. 10, Issue 8, pp. 1–7). https://doi.org/10.35940/ijies.f4263.0810823

AUTHOR PROFILE

Usha Kirana S P is currently working in Don Bosco Institute of Technology, Bangalore, Visvesvaraya Technological University (VTU) Belagavi and working in the area cloud computing and optimization. She has received her Doctorate from VTU Belagvi in 2024 and M.Tech from VTU Belagavi.