International Journal of Mathematical Applications and Solutions
Vol 2, Issue 2 2026

Cloud-Resource Scheduling Framework with Energy Consideration for Autonomic Operations

Rakesh Kumar *, Venkat. P, Vihari

Department of Mathematics,

Mathlogix publications

Article Info

Received: 18-03-2026 **Revised:22-04-2026** **Accepted:04-05-2026** **Published:14-05-2026**

Abstract

The term "cloud computing" refers to an infrastructure that allows for the provision of shared, internet-based services on an as-needed or subscription basis. Cloud services are available for rent from a number of cloud service providers (CSP). Demand for cloud services is on the rise, therefore it's important to make sure the current infrastructure can handle it. But with all those data centers and servers, scaling comes with a hefty price tag in terms of energy usage. The operational expenses are impacted by the unnecessary power usage, which ultimately impacts its consumers. The environment is also impacted by CO2 emissions. Inadequate distribution of resources, such as data centers, virtual machines, and servers, also raises operating expenses. In the long run, this can cause customers to lose interest in the cloud service. Overall, it's necessary to make the most efficient use of the resources. This study suggests using an enhanced Antlion (ALO) algorithm to compute various multi-objective functions in order to determine the best solution for resource usage and allocation. In order to optimize resource usage, the suggested technique is tested in cloud simulation settings, where it calculates energy consumption for varying workload quantities and improves the performance of various multi-objective functions. Experimental findings demonstrate that the proposed framework outperforms both current and previous frameworks.

Keywords- Cost, Cloud computing, Energy, Optimization, Resource scheduling, SLA, QoS.

1. Introduction

According to Preethi et al. (2014), optimizing cloud resources entails a series of steps that make the most efficient use of the resources that are already accessible. By taking into account multi-objective functions, this method enhances the system's efficiency. Power consumption may be improved by making good use of available resources during optimization, which will decrease execution time and cost. Examples of existing optimization algorithms with static behavior include bacterial foraging optimization, ant colony algorithms, genetic algorithms, and particle swarm optimization (Kennedy and Eberhart, 2011). Intelligence and optimization, the future techniques, must be dynamic. Before we get to that, there's self-optimization, which is making the system better by using AI tricks. Finding the best way to provide resources is something it does automatically.

Two techniques that may address the optimization issue are gradient-based and gradient free solvers (Moller, 1993). In order to determine the best possible value of a function, gradient-based solvers take use of derivatives. Even though it's quick, extensively used, and scales well, there are a few drawbacks, such as the high processing cost of the gradient, the need for a smooth continuous function, and the presence of local minima. Because gradient-free algorithms don't need derivatives, they may be used to optimization problems where getting derivatives is challenging (Bose and Pain, 2018). The function may be discrete, discontinuous, or noisy. This

indicates that the gradient-free approach is very adaptable to many kinds of problems. As opposed to gradient-based algorithms, which are much faster, the most popular gradient-free algorithms are an exhaustive search, Nelder-mead (Dennis and Woods, 1987), particle-swarm, simulated annealing, genetic algorithms, and others that draw inspiration from nature. Furthermore, test every conceivable solution and choose the best one; this may only work for a tiny issue, but it's worth trying. Although several evolutionary search optimization approaches have recently been introduced in the cloud, there is still need for improvement in order to find the best answer. According to Dewangan et al. (2018), these goals may be attained by including crucial functions such as cost, energy, performance, execution time, carbon-di-oxide emission rate, power consumption, and quality of service. But up to now, the best answer has been found by taking into account no more than three objective functions. The most precise optimum solution may be obtained by taking into account more than three multi-objective (Tamura and Yamada, 2017) functions while addressing the problem. Energy-Aware Autonomic Resource Scheduling Framework for the Cloud (ESCORT) is a suggested model that incorporates many methods and approaches to address current procedures. It also offers potential enhancements to the existing model.

2. Related Work

A state of art survey has been done to study in the area to find the optimal solution for cloud resource scheduling, which can be categorized as follows:

2.1 Search Based Algorithm

One such search-based method is the Tri-Objective Resource Optimization method (Alex and Yamini, 2017). Its goals are to maximize resource use while minimizing power consumption, costs, and carbon-dioxide emissions. If the new result is better than the existing one, or if the two outcomes are equal, TROA will execute by selecting it from the subset. Problems with the Internet of Things include, according to Choi and Lim (2016), efficiently allocating resources to cloud clients in accordance with SLA requirements and establishing a SLA contract between clients and administration. The SLA is a contract that establishes the quality of service between a customer and a provider. Both a class-construct and a work-based SLA were defined by the inventor. The former estimates QoS in relation to execution measures, while the latter uses measurements of individual occupations. Each time an interim regular victor assurance system is used with a due date constraint, the asset, assignment, and suppliers profit proficiently. pMapper uses bin-pressing computation based on an adjusted version of FFD (First fit decreasing) to place virtual machines on servers in order to expedite cost execution. An ABRA (Auction-Based Resource Co-Allocation) display was used to address the co-designation problem. This display enforces penalty charges on assets that remain unallocated after a barter in order to improve asset utilization. The creator of STPA, Short-Term Planning Algorithm (Banu and Saravanan, 2014), developed a methodology to maximize the efficiency of resource policy subscription costs. The provisioning of time-varying workloads has been facilitated by this decision of how to configure virtual machines. We have found and reserved the ideal quantity of virtual machines. But the pricing scheme is out there; here, however, you may optimize the price by subscribing to virtual machines.

2.2 Evolutionary Based Algorithm

Genetic Algorithm (Sivanandam and Deepa, 2008), this depends on a developmental model for complex frameworks. This calculation is a worldwide, productive and parallel looking system. It resembles a characteristic advancement technique. Hence, GA (Bhunia et al., 2017) is well reasonable for tackling the asset improvement issue. Numerous point looking is an upgraded

highlight of hereditary calculation. The pursuit space is decreased by utilizing GA (Sahoo, 2017) strategies in asset advancement. In this approach, the hopeful's arrangements are spoken to as far as chromosomes, which are retouched in each stage. GA utilizes the choice approach to discover the survival of wellness work. For every last stage, the most elevated wellness esteem is found and those qualities are acquired to straightaway age.

CCGA (Yusoh and Tang, 2012), Evolutionary Algorithm used to streamline cost in this paper. For the composite asset improvement, the writer expected that asset is now running on their VM's and where, the workload of uses and asset limits are changing after some time, the present position should be adjusted. This paper proposed an answer for a disconnected application position issue, to be specific Application Component Placement (ACP) where the area of the information part is thinking about as one of the central factors and consider as preparing correspondence and capacity necessities.

R-P-E Optimization Technique (Qiu et al., 2017), In this paper it is discussed virtualization which is a center innovation of distributed computing underpins cloud suppliers in creating sound asset planning methodologies to diminish control utilization of a physical server. There exists an imperative tradeoff amongst execution and vitality utilization measurements. The fundamental issue is to adjust the confused unwavering quality execution vitality (R-P-E). For a cure, fundamentally examine R-P-E relationship models and comparing asset advancement advances for cloud administrations utilizing retrying and registration blame recuperation systems. The proposed display envelops Markov models, the Laplace-Stieltjes change, a Bayesian approach and a recursive strategy.

OCRP (Chaisiri et al., 2012), the OCRP algorithm is proposed to limit the aggregate cost for provisioning assets in a specific time. A heuristic technique for benefit reservation and K-closest neighbor's calculation was connected to anticipate the request of assets. Ideal virtual machine position (OVMP) calculation can yield the ideal answer for both asset provisioning and VM situation in two provisioning stages.

2.3 Autonomic Algorithm

Soccer (Singh et al., 2016), it is an automatic optimization technique, which is based on energy optimization for cloud resources in data centers. It considers the energy parameter as a QoS and reduces the energy consumption automatic.

GRASP (de C. Coutinho et al., 2014), in this paper it is discussed control productive plans for reserving, question preparing, and warm administration are required because of the expanding measure of waste warmth that server farms disseminate for application administrations. Moreover, they have estimated, how many shoppers can utilize the transfer speed, CPU, and memory at any given time for the duration of the day. Application schedulers have diverse strategies shift as per the goal work: limit add up to execution time, limit installment cost to execute, limit the interest for control while keeping up an administration level certification for the shoppers, adjust the heap on assets utilized while meeting the due date requirements of the application et cetera. Asset allotment issues in mists are NP-difficult issues, so there are no productive strategies to unravel them. Concerning the cloud customer point of view, registering assets can be obtained by reservation or on-request designs. To explain it, a number programming definition and a heuristic in light of Greedy Randomized Adaptive Search Procedure (GRASP). CHOPPER (Gill et al., 2017) proposed autonomic resource management with self-characteristics to optimize energy and cost.

2.4 Nature Based Algorithms

Similar to how ants find food, the Ant Colony Optimization Algorithm is a randomized probabilistic technique that finds the optimal path (Al Salami, 2009). Ants often release pheromones when they find a path. More ants, working in tandem to find the best path, draw this one in. In a distributed computing environment, a cloud service provider makes available a variety of cloud resources across several virtual machines. When it comes to optimizing cloud assets, ACO plays a crucial role. Assets are like food, virtual machines are like nodes, and ants are like operators. The insect travels between the hubs to distribute the requested tasks to the resources in the cloud.

An algorithm for optimizing particle swarms that is part of the super arrangement of swarm intelligence (Chen and Yu, 2005). An problem is improved in this computation by using a self-adaptive iterative technique. A predetermined metric of value is repeatedly considered when handling the optimistic configurations. 'Particles' are the shorthand for the applicant arrangement population in PSO. Particle swarm optimization (PSO) incorporates these particles' evolution in the pursuit space. Simple numerical formulas controlling the molecule's velocity and location direct the evolution. Every molecule in close proximity to the best-known location affects its growth. Besides this, it is also directed toward the most popular spots in the hunting area, which are updated as other particles discover better spots. Thanks to these molecular advancements, the swarm should zero in on the optimal configuration. Particle swarm optimization is a metaheuristic algorithm, much as ant state optimization. The Swarm is a population of the optimistic configurations that a fundamental sort of computation needs to function. Each rival configuration is extremely similar to a molecule inside the system. The framework ought to serve as the area for layout.

The algorithm relies on hyper-heuristics approach for optimizing bacterial foraging (Das et al., 2009). A bacteria and its movement are used as heuristics in this computation to represent the partial result. Chemical taxis, swarming, proliferation, end, and dispersion are all steps in the process that lead to better bacterial scavenging calculations. To increase the power consumption per time unit (P/T), the computation is based on how the creature locates the food. Table 1 displays the results of a state-of-the-art comparative study of several optimization strategies; Figure 1 classifies these techniques into four groups: search-based, evolutionary-based, nature-based, and autonomic-based.

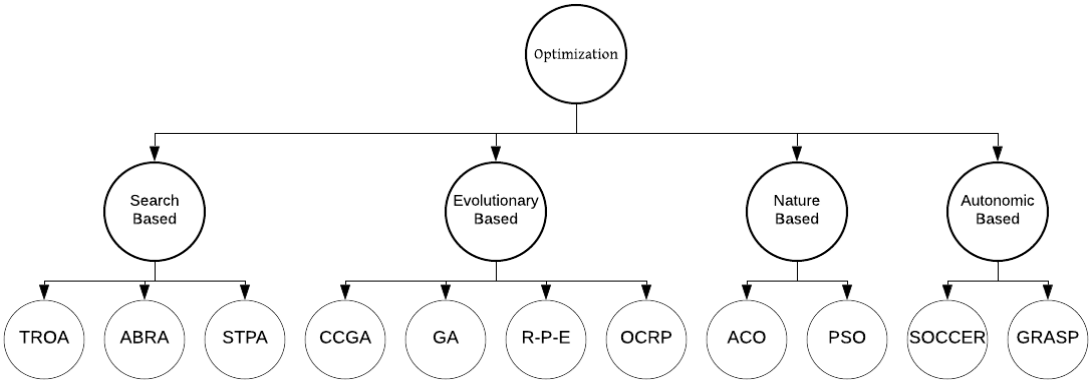


Figure 1. Optimization techniques in cloud

The comparison has been carried out based on multi-objective functions.

Table 1. Comparative analysis of different optimization techniques with ESCORT

Performance Metrics	Optimization Techniques								
	TROA	SOCCER	STPA	ABRA	CCGA	OCRP	R-P-E	GRASP	ESCORT
SLA-VR	**No**	**No**	**No**	**Yes**	**No**	**No**	**No**	**No**	**Yes**
QoS	**No**	**No**	**Yes**	**Yes**	**No**	**No**	**No**	**No**	**Yes**
Execution Time	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**Yes**	**Yes**
Cost	**Yes**	**Yes**	**No**	**Yes**	**Yes**	**Yes**	**No**	**Yes**	**Yes**
Energy	**No**	**Yes**	**Yes**	**No**	**No**	**No**	**No**	**Yes**	**Yes**
Power	**Yes**	**No**	**No**	**No**	**No**	**No**	**Yes**	**No**	**No**
CO2 Emission	**Yes**	**No**	**No**	**No**	**No**	**No**	**No**	**No**	**No**

3. Limitations and Possible Improvements

The resources may be handled using multi-objective functions and sent to a pool. Then, they can be allocated according to the quality of service needs. According to the aforementioned research, in order to get the most out of cloud resources, researchers optimized for price, power, execution time, and CO_2 emission rate. Using the Antlion optimization algorithm—which has been used before for network optimization—to optimize energy consumption in order to decrease execution cost, time, SLA violation rate, and maximize resource usage is the goal of this research. In order to improve energy usage and other multi-objective functions, this work presents a unique approach to cloud computing resource scheduling using the Antlion optimization algorithm. Table 2 displays the many names used in this article, ranging from equations 1 to 13.

Table 2. Notations

Notation	**Definition**
n	number of workloads
M	number of variables
R_i	failure rate
SV_R	SLA violation rate
w_i	the weight of every SLA
RC_i	request workload completion time
RS_i	workload submission time
SLA_Cost	cost of SLA
M_{Ant}	the position of each ant store in the matrix
M_{OA}	matrix for saving the fitness of each ant
$M_{Antlion}$	is saving the position of each antlion
M_{OAL}	the matrix to store the fitness of each antlion
$R(e)$	random walk
C_s	cumulative sum
e_n	number of iteration
E	step size
$r(e)$	random function
$F1, F2, F3$	objective functions
o_i, p_i	minimum and maximum of the random walk of i-th variable respectively

4. Methodology

4.1 Formal Optimization Model

The flow of formal optimization model is representing through Figure 2 and the architecture of the proposed model ant-lion based model is mentioned in Figure 3. In general, to obtain the optimal solution, population and fitness values has to be calculated. On the basis of these values if the method meets the optimization criteria then it will stop to search the solution and resources will be sent to the pool else the process need to be regenerate the population and calculate the fitness values.

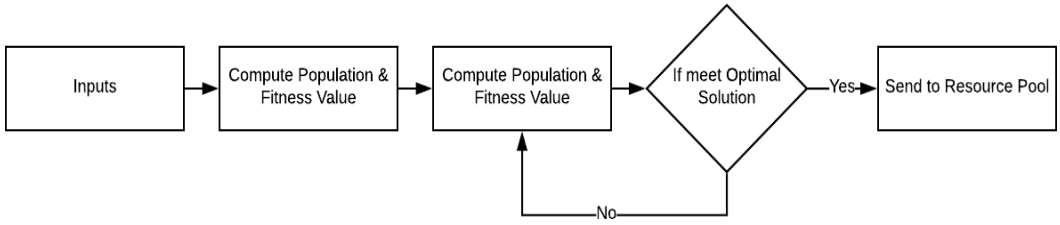


Figure 2. Flow of formal optimization model

4.2 Architecture of Antlion Based Optimization Model

The architecture of proposed Antlion based optimization architecture is presenting in Figure 3, where cloud users are submitting the workload to the task manager for scheduling the resources.

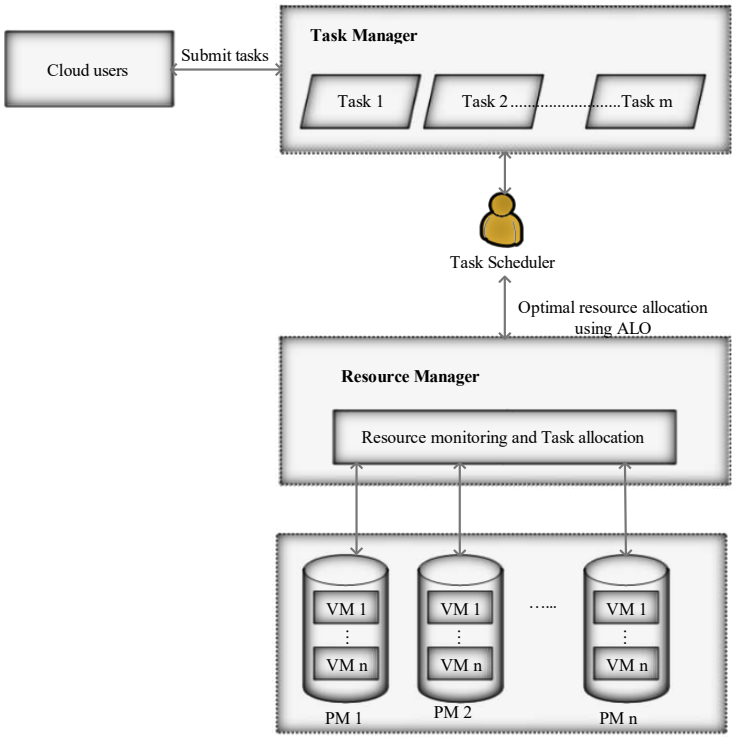


Figure 3. Proposed antlion based optimization model ESCORT

4.3 Resource Management

Resources will be managing through the calculation of performance metrics like failure rate, cost, energy, and SLA violation rate.

The failure rate of resources is a failure rate of every workload divisible by a number of workloads.

$$\text{Failure rate} = \frac{failure(R_i)}{n} \tag{1}$$

SLA violation rate is calculating by multiplication of failure rate and weight of every SLA.

$$SVR = failure\,rate * \sum_{i=1}^{n} w_i \tag{2}$$

Execution time is the difference between workload completion time and workload submission time.

$$Executiontime = \sum_{i=1}^{n} \left(\frac{RC_i - RS_i}{n}\right) \tag{3}$$

SLA cost is the cost of SLA violation rate and execution time. If both are high, then the cost will increase and vice versa.

$$SLA_Cost = SV_R * \sum_{i=1}^{n} Executiontime \tag{4}$$

4.4 Resource Allocation

In this paper for the best VM selection used ant lion optimization (ALO). After the centroid positions are finished in the grouping procedure, consider the hubs which are at the closest separation and furthermore the following closest separation from the centroid by utilizing ALO optimization. In this method, the initialization is done by using a random selection of antlions and ants (VM). After that, the multi-objectives (resource utilization, energy, and cost) is calculated between each VMs, from that the fitness value of ant and ant lion is calculated and ant lion fitness value sorted to find the best ant lion is named as elite antlion (best VM).

Ant's VMs are moved at arbitrarily to look for their nourishment. An arbitrary walk is utilized to demonstrate their developments utilizing total entirety work and an irregular capacity connected for an alternate emphasis. The random walk of ant can be calculated by equation (5)

$$R(e) = [0, C_S(2r(e_1) - 1), C_s(2r(e_2) - 1), ... C_s(2r(e_n) - 1)] \tag{5}$$

$r(e) \rightarrow$ is an arbitrary capacity takes the esteem '1' if the irregular number produced is more prominent than 0.5 else it is set to '0'.

To keep ant arbitrary strolls inside the hunting space to refresh their situation concerning chosen subterranean insect lion utilized min-max standardization. Calculate the min-max normalization by using the equation

$$R_i^e = \{[(R_i^e - o_i)(R_i^e - q_i^e)]/(p_i^e - o_i)\} + q_i] \tag{6}$$

To compute fitness value, consider that the ants are caught in any of the haphazardly chose antlion. The roulette wheel is utilized to choose the best subterranean insect lion from its populace in view of wellness count. This very fit insect lion to get the ants with high likelihood.

$$VM_U = VM.CPU_U + VM.BW_U + VM.RAM_U \quad (7)$$

Therefore, the first objective function is,

$$F_1 = \max\{VM_u\} \quad (8)$$

Computation of Energy consumption will be done through

$$S = k * \max + (1-k) * \max * VM_u \quad (9)$$

Therefor the second objective is,

$$F_2 = \min\{s\} \quad (10)$$

And the cost of the resource is,

$$Cost\, C = c * S \quad (11)$$

Where c is the cost of 1 kW power.

Therefor the third objective function is,

$$F_3 = \min\{C\} \quad (12)$$

Antlions are building their trap with their wellness esteem. On the off chance that the subterranean insect is inside the trap, at that point it shoots the sand outside the focal point of the pit. It influences the subterranean insect to slide down when it attempts to escape from the antlion.

Solution updating, ant achieves the base of the pit, and after that the antlion pulls it in the sand and eats its body. Next, the antlion adjusts its position and fabricate another trap to get another prey.

Antlion gets an insect, when the wellness of antis more noteworthy than the wellness of antlion. It is given by,

$$AL^t_j = A^t_i \text{ If } fitness(A^t_i) > fitness(AL^t_j) \quad (13)$$

The algorithm of the above method is presenting in the next section.

5. Implementation Results and Analysis

The proposed model ESCORT has been implemented and simulated in clouds. In this simulation, 100 workloads are submitted to the system, and 20 VM's are available in the resource pool to submitted workloads. The workloads are clustered through k-mean clustering algorism and it is clustered to four centroid values C1, C2, C3, and C4. The workloads are sorted and arranged according to user priority, i.e. execution time.

5.1 Computing Energy Consumption

The energy consumption rate for each workload has been computed and taken an average of each iteration. The outcome of this experiment is presenting in Table 3.

The common workloads are computed in existing frameworks, SOCCER and CHOPPER and observed that when the workloads are increasing the energy consumption rate are also increasing as mentioned in Figure 4.

Table 3. Energy consumption rate (KWh)

Number of workloads	200	400	600	800	1000
ESCORT	9.7 kw	13.5 kw	20 kw	27kw	37 kw
CHOPPER	12 kw	15 kw	26.5 kw	32 kw	41.3 kw
SOCCER	10.5 kw	14.7 kw	23 kw	29.4 kw	38.5 kw

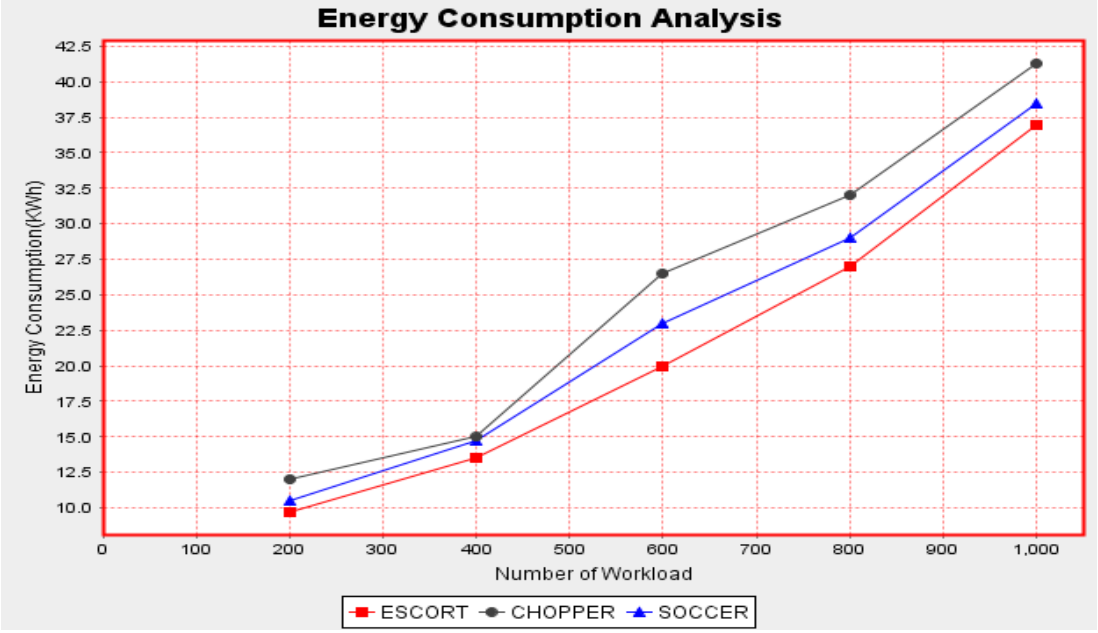


Figure 4. Energy consumption analysis

The above results are evidence that the proposed method is better in optimization when it compared with energy consumption rate with existing models. Cloud user submitted 200 workloads to allocate 20 VM's as mentioned in Figure 4 and recorded the consumption rate for 400, 600, 800, and 1000 workloads. ESCORT is cost-effective, produce less SLA violation rate and consumes less energy as compared to existing methods.

5.2 Computing SLA Violation Rate

SLA violation rate and Execution time for each workload have been computed and taken an average of each iteration. The common workload dataset is computed in ESCORT, SOCCER and CHOPPER, recorded as mentioned in Table 4, when the execution time of workloads is increasing the SLA violation rate are also increasing.

Table 4. SLA violation rate

Number of workloads	10	20	30	40	50	60
ESCORT	200	250	300	450	650	695
CHOPPER	300	350	500	680	800	900
SOCCER	250	300	400	620	750	870

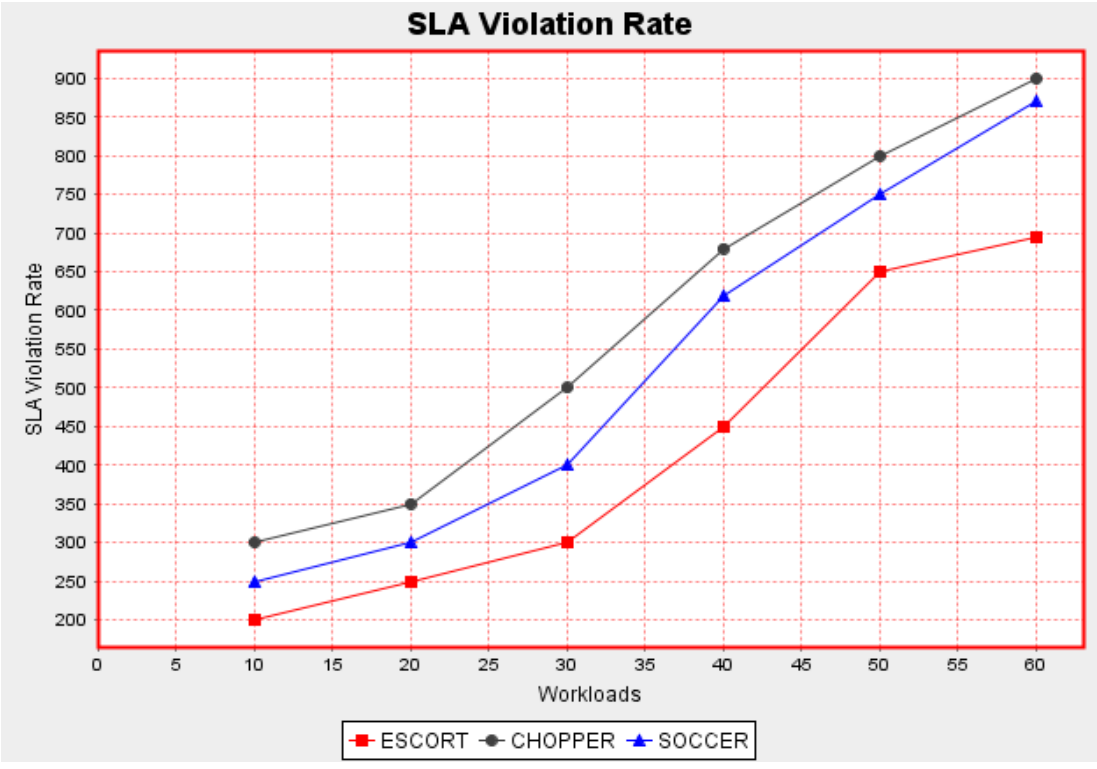


Figure 5. SLA violation rate

The average values of SLA violation rate have been recorded for the proposed method and the other two existing methods. When the number of workloads is increasing the SLA violation rate are also increasing. The graphical representation of different methods is mentioned in Figure 5.

5.3 Computing Execution Cost

SLA violation rate and Execution cost for each workload have been computed and taken an average of each iteration. The same workloads are also computed in existing frameworks, SOCCER and CHOPPER. When the execution cost of workloads is increasing the SLA violation rate are also increasing as results recorded in Table 5.

Table 5. Execution cost ($)

Number of workloads	0-10	10-20	20-30	30-40	40-50	50-60
ESCORT	18	20	25	38	40	55
CHOPPER	30	35	48	58	63	75
SOCCER	25	30	40	50	60	70

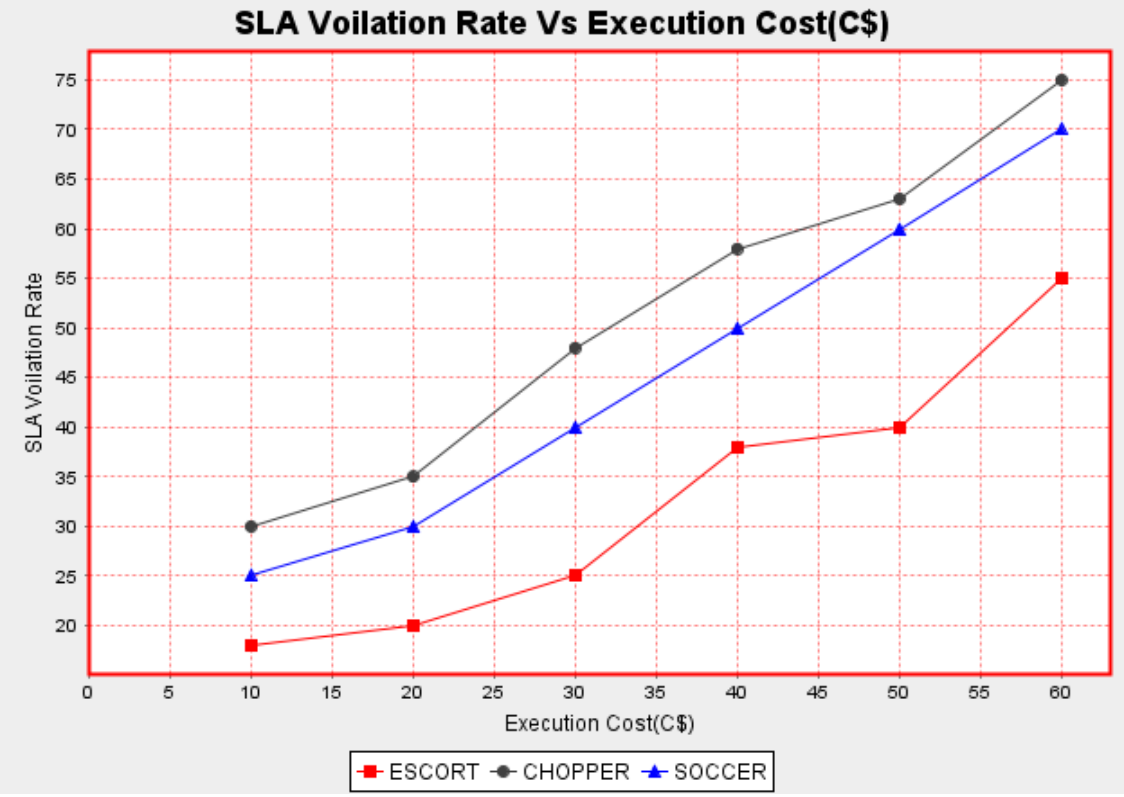


Figure 6. Execution cost ($)

The average values of execution cost have been recorded for the proposed method and the other two existing methods. The graphical representation of different methods is mentioned in Figure 6.

5.4 Comparative Analysis

The average of different multi-objectives performance metrics has recorded in Table 6. Overall evaluation of ESCORT founds that, it producing efficient results as compared to existing frameworks.

Table 6. Analysis of performance metrics (average)

Frameworks	SLA Violation Rate (Workload 10 to 60)	Execution Cost (Workload 10 to 60)	Energy Consumption (Workloads 200 to 1000)
ESCORT	424.16	$ 32.66	21.44 KW
CHOPPER	588.33	$ 51.5	25.36 KW
SOCCER	531.66	$ 45.83	23.14 KW

6. Conclusion

Energy usage, execution cost, and SLA violation rate optimization is discussed in this study using the ESCORT framework. Within the cloudsim environment, this framework is both implemented and simulated. When it's put into action in the actual cloud, the outcome can be different. This document lays out the steps necessary to implement ESCORT, which will allow you to optimize your cloud scheduling performance while reducing your energy usage, SLA violation rate, execution time, and costs. With an average consumption of 21.44 KW, ESCORT outperforms the other examined approaches, which use 23.14 KW and 25.36 KW of energy, according to the trial data. To make sure ESCORT works as well as it can, we compared its planned work with alternative methods. The outcomes of the ESCORT simulation aim to optimize resource usage while minimizing the execution cost and the number of SLA violations. One area that might be improved upon is the lack of screening algorithms for malicious activity in this work. We plan to address this in future iterations.

References

The ant colony optimization technique was developed by Al Salami in 2009. Public Computing and Communications, vol. 4, no. 3, pp. 823–826.
Authors: Alex (2017) and Yamini R. Algorithms for optimizing cloud resources compared. Journal of Pure and Applied Mathematics: International, 16(21), 847-855.
In 2014, Banu and Saravanan published a paper. Improving the Insurance as a Service (IaaS) cloud's cost-subscription policy.
Volume 6, Issue 5, Pages 296-301, International Journal of Engineering Trends and Technology.
Sahoo, L., Bhunia, A. K., & Duary, A. (2017). Optimization of dependability and redundancy in a series system with multiple-choice inputs using a hybrid technique based on genetic algorithms. Journal of Mathematical, Engineering, and Management Sciences: An International Journal, 2(3), 185-212.
Published in 2018 by Bose and Pain. Metaheuristic Approach of Multi-Objective Optimization during EDM Process. Journal of Mathematical, Engineering, and Management Sciences: An International Journal, Volume 3, Issue 3, Pages 301-314.
Written by Chaisiri, S., Lee, B., and Niyato, D. (2012). Efficiency improvement of cloud computing resource provisioning costs.
5(2), 164-177, IEEE Transactions on Services Computing.

G. C. Chen and J. S. Yu (2005). Optimization method based on particle swarms. In their 2016 article "Information and Control" (vol. 34, no. 3), Choi and Lim discuss. Allocation of cloud computing resources for the Internet of Things: an optimization strategy.
Volume 12, Issue 3, page 3479247, International Journal of Distributed Sensor Networks.
A study conducted by Das, Biswas, Dasgupta, and Abraham in 2009 was published. An method for optimizing bacterial foraging: theory, analysis, and practical applications. Pages 23–55 of Volume 3 of Foundations of Computational Intelligence. Berlin and Heidelberg: Springer.
Y. Frota, L. M. Drummond, and R. De C. Coutinho (2014). Optimization of a cloud resource management issue from a customer viewpoint. Parallel Processing Workshops at Euro-Par 2013, pages 218–227.
In 1987, Dennis and Woods published a paper. Numerical optimization using microprocessors: the Nelder-Mead simplex method. This article is part of the SIAM Philadelphia conference on "New Computing Environments: Microcomputers in Large-Scale Computing" (pp. 6-122).
In 2018, Dewangan, B. K., Agarwal, A., Venkatadri, and Pasricha published a paper. Comparison of cloud resource scheduling methods. Citation: International Journal of Computer Sciences and Engineering, volume 6, issue 8, pages 167–173.
In (2017), Gill, S. S., Chana, I., Singh, M., and Buyya, R. were authors. CHOPPER is a smart autonomic resource management system for the cloud that takes quality of service into account. Publication: Cluster Computing, Volume 21, Issue 2, Pages 1203–1241.
In 2011, Kennedy and Eberhart published a book. Encyclopedia of Machine Learning article on particle swarm optimization (pages 760–766). Publishing House: Springer, Boston, MA.
According to Moller (1993). Accelerated supervised learning using a scaled conjugate gradient technique. Neuronal networks, volume 6, issue 4, pages 525-533.
The authors of this work are Preethi (B.), Kamalanathan (C.), Ramesh (S. M.), Shanamathi (S.), and Bama (P. S.). Using efficient load balancing methods, cloud computing resources are optimized. Page 1 of the International Advanced Research Journal in Science, Engineering and Technology says it all. 20 to 22.
In 2017, Qiu, X., Dai, Y., Xiang, Y., and Xing, L. published. Optimization of resources and correlation modeling for fault-tolerant cloud services. IEEE Transactions on Cloud Computing, Volume 5, Issue 1, Pages 1–13.
By Sahoo, L. in 2017. Using a genetic algorithm to solve redundancy and reliability allocation issues in a fuzzy setting. Journal of Mathematical, Engineering, and Management Sciences: An International Journal, 2(4), 259-272.
In 2016, Singh, S., Chana, I., Singh, M., and Buyya, R. published one piece. Optimizing energy-efficient cloud resources autonomously: SOCCER. The article may be found in Cluster Computing, volume 19, issue 4, pages 1787–1800.

Publication year: 2008 by Sivanandam and Deepa. Concerns in optimizing genetic algorithms. Chapter 1: Genetic Algorithms and Its Applications (pages 165-209). Berlin and Heidelberg: Springer.
Publication: 2017 by Tamura and Yamada. A reliability analysis tool for big data and cloud computing based on a multi-dimensional stochastic noise model. Volume 2, Issue 4, Pages 273-287, International Journal of Mathematical, Engineering and Management Sciences.
Z. I. M. Yusoh and M. Tang published in June 2012. Using evolutionary methods, we optimize cloud computing resources and install composite SaaS applications. In 2012 IEEE 5th International Conference on Cloud Computing (CLOUD), IEEE (pp. 590-597).