Cost-Aware Ant Colony Optimization Based Model for Load Balancing in Cloud Computing

Malini Alagarsamy[1], Ajitha Sundarji[2], Aparna Arunachalapandi[3], and Keerthanaa Kalyanasundaram[4]
[1]Department of Computer Science, Thiagarajar College of Engineering, India
[2]AstraZeneca GTC, India
[3]Wipro Technologies Limited, India
[4]HCL Technologies, India

Abstract: *Balancing the incoming data traffic across the servers is termed as Load balancing. In cloud computing, Load balancing means distributing loads across the cloud infrastructure. The performance of cloud computing depends on the different factors which include balancing the loads at the data center which increase the server utilization. Proper utilization of resources is termed as server utilization. The power consumption decreases with an increase in server utilization which in turn reduces the carbon footprint of the virtual machines at the data center. In this paper, the cost-aware ant colony optimization based load balancing model is proposed to minimize the execution time, response time and cost in a dynamic environment. This model enables to balance the load across the virtual machines in the data center and evaluate the overall performance with various load balancing models. As an average, the proposed model reduces carbon footprint by 45% than existing methods.*

Keywords: *Scheduling algorithms, application virtualization, power, energy.*

Received August 3, 2020; accepted April 7, 2021
https://doi.org/10.34028/iajit/18/5/12

1. Introduction

Cloud Computing is a buzzing technology in today's internet world. Cloud enables flexible usage of data and resources. Cloud users can conveniently store and access data anytime anywhere. Its ubiquitous nature has resulted in the increasing number of users in this domain. As per National Institute of Standards and Technology, "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [15]. Cloud eases the feature of data sharing for the internet users. The availability and scalability feature of cloud has resulted in avoiding the investments of larger capital costs for companies thereby attracting the users in a large scale.

For the betterment of our environment, the idea of green computing has emerged. Green computing is a technique of efficiently utilizing Information Technology (IT) resources by applying better policies and algorithms. One of the important key concepts of green computing is virtualization [4]. Virtualization is generally referred as the abstraction of computing resources such as Central Processing Unit (CPU), memory, network, storage and database related applications and the clients utilizing the service provisioned by the cloud providers [11]. It enables multi- tenancy model by providing a scalable, shared resource platform for all tenants [27].

Besides the advantages of virtualization in cloud computing, one of the environmental issues is increase in carbon footprint. Carbon footprint is the amount of carbon dioxide emission in the environment. In context of cloud, this issue is faced due to inefficient usage of data center. The data center's efficiency could be increased by applying appropriate load balancing algorithms. By balancing the workload across all the nodes of a cloud, overheating is reduced which in turn reduces the energy consumption [26]. As the energy consumption increases, carbon footprint increases. Therefore, by reducing the energy consumption, the aim of green computing is achieved [24].

In this paper, Ant Colony Optimization (ACO) based carbon aware load balancing model is proposed for improving the allocation of tasks to the virtual machines. This model also provides the comparison with various load balancing algorithms. The objective of the proposed work is

1. To reduce the processing time.
2. To lessen the response time.
3. To lower the cost.
4. To bring down the power consumption.
5. To minimize the carbon footprint.

The overall motivation is to optimize the approach of Load balancing in cloud computing. The load balancing algorithms are analyzed under different service broker policies in order to provide better performance analysis.

This paper is organized as follows: section 2 shows some of the existing load balancing algorithms, section 3 discusses the proposed ACO based carbon aware load balancing model and its overall design, section 4 deals with the real time implementation and results and section 6 presents the conclusion.

2. Literature Survey

Naqvi *et al*. [17] has proposed a bio-inspired algorithm called Ant Colony optimization for load balancing. The ant colony optimization algorithm is the swarm based genetic algorithm [16]. It works on the mechanism of real ants using pheromones in order to explore its path. In the same way, the allocation path of the cloudlets is identified based on the minimal path cost in a probabilistic manner.

Subalakshmi and Malarvizhi [23] suggested the algorithm called Equally Spread current execution load Algorithm. The Equally Spread current execution load algorithm balances the load across the virtual machines. It maintains the index table that contains the number of requests and currently allocated virtual machines. When a new request arises, the least loaded virtual machine is identified by referring to the index table. If there is more than one least loaded virtual machine, then the first identified virtual machine is chosen.

Kushwaha and Gupta [12] proposed the algorithm called Round-robin load balancing algorithm. The Round-robin load balancing algorithm uses the time quantum for the allocation of virtual machines. The first virtual machine is selected randomly and then further allocations are made in a circular manner based on the time quantum. The major drawback of this algorithm is that the task is made to wait for a long time in the queue if the virtual machine is not available thereby increasing the execution time.

Kashyap and Viradiya [8] suggested the genetic algorithm called Honey bee load balancing algorithm. The Honey bee algorithm is the bio-inspired algorithm based on the behavior of the honey bee. It keeps tracking the workload of each virtual machine. The task from the overloaded virtual machine is removed and is given higher priority so that it can be assigned to the next lightly loaded machines.

Nitika *et al*. [18] has implemented the algorithm called throttled load balancing algorithm. The throttled algorithm maintains the index table that contains the virtual machine and its state. When a new request arises it checks for the availability of the virtual machine by referring the index table. If all the virtual machines are active then the requests are queued until the virtual machine becomes available.

The proposed load balancing model is done with the FaceBook data set available from [3] and energy-aware simulator CloudAnalyst [28]. The CloudAnalyst is an extension of CloudSim toolkit with an additional Graphical User Interface which allows the user to configure the cloud environment in detail and also enables the user to experiment with the large scale cloud environment easily [2].

3. Proposed Method

The ACO based carbon aware model identifies a solution for efficient load balancing by considering factors such as processing time, response time to reduce carbon footprint in the cloud computing environment. The Proposed algorithm is based on genetic algorithm Ant colony optimization which uses path cost and threshold. The major components are

1. User Base.
2. Data center selector.
3. Virtual Machine (VM) selector and allocator.
4. Efficiency analyzer as shown in Figure 1.

3.1. UserBase

The User base is a collection of users grouped under a region. The chosen configuration includes 6 regions across the world [13]. A Single-user base may consist of thousands of users and each user, in turn, may request for thousands of tasks. The user bases generate traffic as in real time. The increasing number of users determines the efficiency of the simulation. The number of simultaneous users in each user base can be bundled as a single unit by grouping factor.

3.2. Data Center Selector

The data center Selector maps the data center with the traffic generating user base depending on the service broker policy. The service broker policies are the Closest Data Centre (CDC) and Optimize Response Time (ORT).

3.2.1. Closest Data Center (CDC)

The user bases are routed to the data center which has the minimum network latency irrespective of network bandwidth [19]. If there are two data centers under the same region proximity one of them are randomly chosen. This policy calls the data center selector to identify the closest data center. This default broker policy is advantageous in case of the requests are being processed by the data center within the same location.

3.2.2. Optimise Response Time (ORT)

This service broker policy uses the same methodology as the closest data center policy to select the data center as per the network latency [10]. In addition, it calculates the current response time and checks whether the estimated response time is the same as that of the closest data center. Otherwise, the data center with the least response time or that within the closest proximity is chosen evenly with the occurrence ratio of 50:50.

3.3. VM Selector and Allocator

VM selector and allocator use the VM load balancer to allocate the cloudlets (user requested tasks) to the Virtual Machine. The existing load balancing policies in cloud analyst are Round-Robin, Equally Spread current execution load, Throttled, Honey bee, Ant colony optimization.

3.3.1. Cost-aware ACO Based Model for Load Balancing

The proposed Cost aware Ant Colony Optimization (CACO) algorithm uses the approach of swarm-based Ant colony optimization algorithm as shown in Code Algorithm 1. This algorithm reduces the power consumption thereby minimizing the outage probability and the performance is depicted in Figure 7. Ants deposit a type of biochemical substance known as a pheromone in order to explore its path. Similarly, ants are considered as cloudlets. Each cloudlet maintains the pheromones table in which path cost is updated. Initially, each cloudlet chooses the virtual machine randomly. The next available virtual machine is identified based on the score function and workload. After completing its tour, update the pheromones table. If all the cloudlets completed their trips then calculate the make span of the cloudlets and retain the optimal solution. If it reaches the maximum limit of iteration then stops the iteration and yield the best solution.

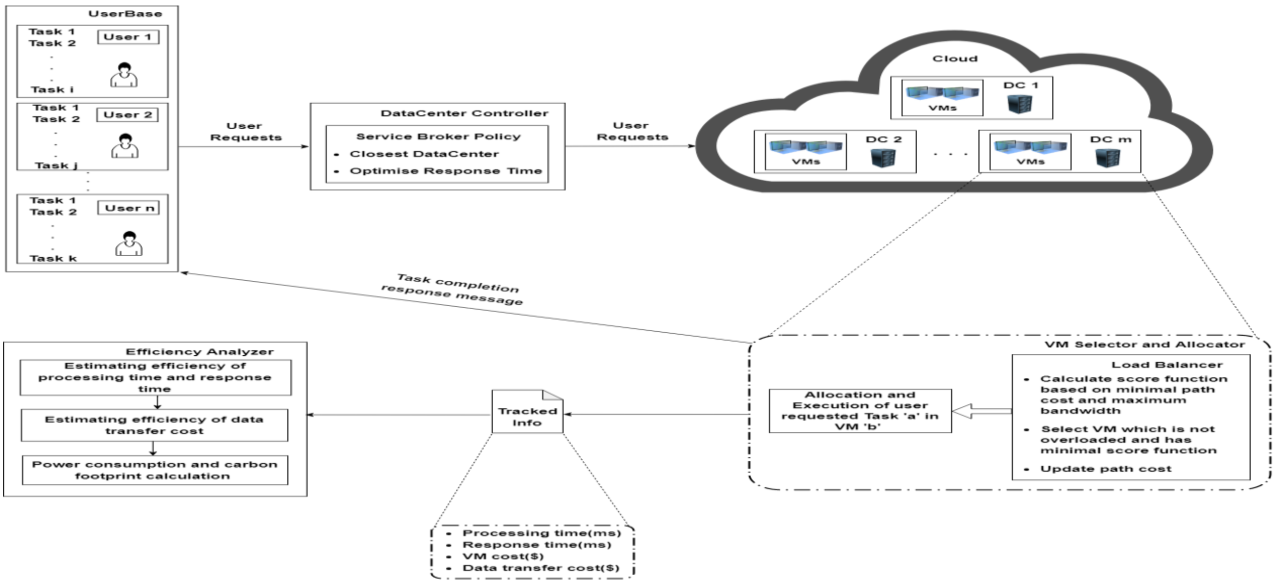


Figure 1. CACO load balancing model.

Algorithm 1: Cost aware Ant Colony Optimization (CACO) Load Balancing Algorithm

Input: List of ants and VM's
Output: Allocation of ants to the VM's
Steps:
Initialize VM's state and count
Initialize Pheromones table
Set the upper and lower threshold values
Initialize under Loaded and Over Loaded queues
Get next AvailableVM()
{
Position each ant in a virtual machine randomly
While (every ant has not build a solution) do
For each ant do
Choose VM for next task by:
Min (Path cost+ ((Max BW-current BW)/Max BW) + VM cost+ Memory Cost)
If chosen VM is overloaded then choose the VM from the underloaded Queue
Update the count of ants assigned to the chosen VM
Initialize under Loaded and Over Loaded queues
End for
End while
Update the pheromone
}

4. Experimental Results and Discussions

The purpose of our model is to reduce the carbon footprint by efficiently allocating the tasks to the virtual machines by using different service broker policies. Social networking media connect people across the world in the online platform. It is one of the largest internet applications that can be satisfied via cloud computing. FaceBook is one such social media application with a large population of users. As of 30 June 2017, FaceBook has 1.97 million users across various geographic locations as shown in Table 1. In our experimental model, we have used CloudAnalyst tool for the simulation to analyze the characteristics of FaceBook application in the cloud environment.

Table 1. FaceBook subscribers' statistics as of June 30, 2017 (internetworldstats, 2018).

World Regions	FaceBook 30 June 2017
Africa	160,207,000
Asia	736,003,000
Europe	343,273,740
Latin America/Caribbean	370,975,340
Middle East	86,700,000
North America	263,081,200
Oceania/Australia	19,463,250
World Total	1,979,703,530

4.1. Experimental Setup

We have chosen six user bases to indicate the various geographic locations as shown in Table 2. For the ease of simulation, we have chosen 1/10th of FaceBook's population. The peak utilization time is assumed to be at night for about 2 hours which is considered as the time zone. It is also assumed that 1% of users are using the platform simultaneously during the peak hours and a single tenth of them are using the platform simultaneously during the off-peak hours. The CloudAnalyst tool enables various parameters as input as mentioned in Table 3.

Table 2. User configurations used in the experiments.

User Base	Region	Peak Hours (GMT)	Simultaneous Online Users During Peak Hours	Simultaneous Online Users During Off-peak Hours
N. America	0	13:00–15:00	2630812	263081
S. America	1	15:00–17:00	3709753	370975
Europe	2	20:00–2:00	3432737	343273
Asia	3	01:00–3:00	7360030	736003
Africa	4	21:00–23:00	1602070	160207
Oceania/Australia	5	09:00–11:00	194632	19463

Table 3. Data centre parameter values used in the experiments.

Parameters		Values assigned
Virtual Machine (VM)	Number of VMs	Based on the Scenario
	Image size	10,000
	Memory	512 MB
	Bandwidth	1000 MB
Data centre	Region	Based on the scenario
	Architecture	x86
	Operating system	Linux
	Virtual Machine Monitor (VMM)	Xen
	Memory per machine	4 GB
	Storage per machine	100 TB
	Available bandwidth per machine	1000000 MB
	Number of processors	4
	Processor Speed	10000 MB
	VM Policy	Time shared
User grouping factor in user bases		10000
Request grouping factor in data centres		1000
Executable instruction length		250

4.1.1. Simulation Scenarios

We have used Cloud Analyst simulation tool to analyze the performance of the FaceBook application in cloud environment under two service broker policies:

1. Closest Data Center (CDC).
2. Optimize Response Time (ORT) for six load balancing algorithms including ACO based carbon aware algorithm under different scenarios [14] as mentioned in Table 4.

Table 4. Scenarios chosen for simulation.

Scenario ID	Scenario Configurations
S1	One data centre with 25 VMs, located at region 0
S2	Two data centres with 25 VMs each, located at region 0 and 2 respectively
S3	Two data centres with 25,50 VMs, located at region 0 and 2 respectively.
S4	Three data centres with 25,30, 50 VMs, located at three different regions 0, 2, and 1 respectively

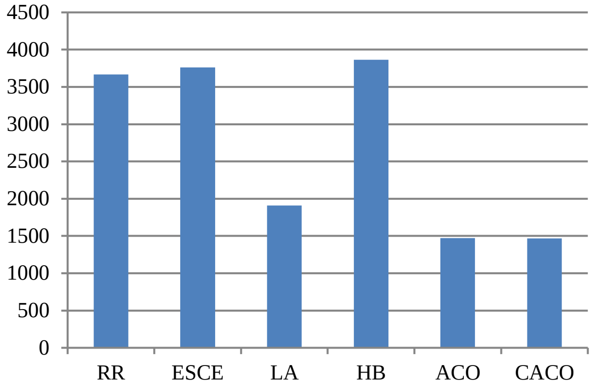
In Scenario S1, simulation is executed with single data center with 25 VMs located in region 0. In the second scenario S2, two data centers are chosen with same count of VMs at different locations. In third scenario S3, two data centers are chosen with variable count of VMs at different geographical locations. In fourth scenario, three data centers with variable count of VMs are chosen at different geographical locations across the globe. The chosen algorithms along with the proposed one are simulated in all the four scenarios and their respective behaviors are analyzed.

4.2. Results

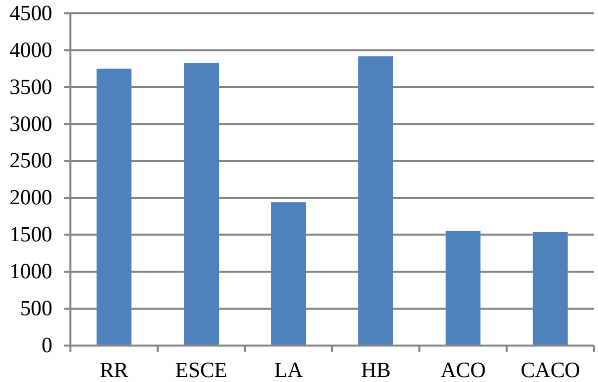
The Min, Max, and overall average values of processing time and response time are recorded for each scenario during the simulation for about 60 hours as shown in Tables 5 and 6 respectively. The existing algorithms such as Round Robin (RR), Equally Spread Current Execution Load (ESCE), Location Aware (LA), Honeybee (HB), ACO and the proposed CACO Algorithm are taken into consideration for analysis. The results are recorded for the factors including total cost and power consumed, energy consumed and carbon footprint which are tabulated in Tables 7, 8, 9, and 10 respectively.

4.2.1. Processing Time

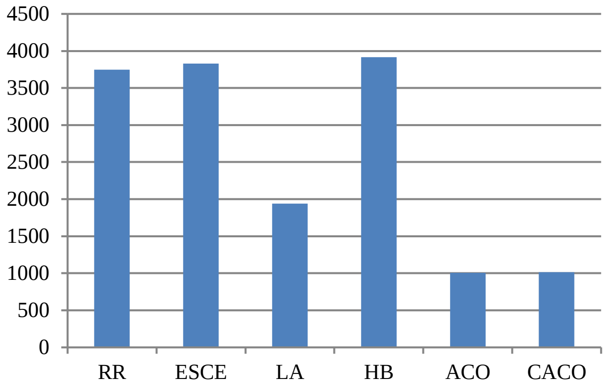
Processing time is calculated based on the length of the tasks requested by users and the capacity of the virtual machines which are assigned to handle the respective tasks [21]. The processing time resulted from the algorithms are represented in milliseconds (ms). The graphical representation of processing time under various scenarios for CDC and ORT are shown in Figures 2 and 3 respectively.



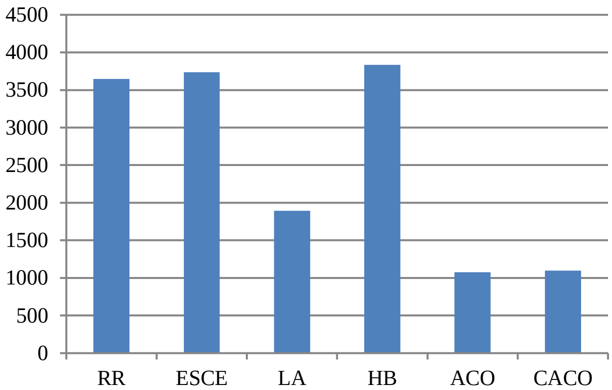
a) S1: One data center with 25 VMs, located at region 0.



b) S2: Two data centers with 25 VMs each, located at region 0 and 2 respectively.

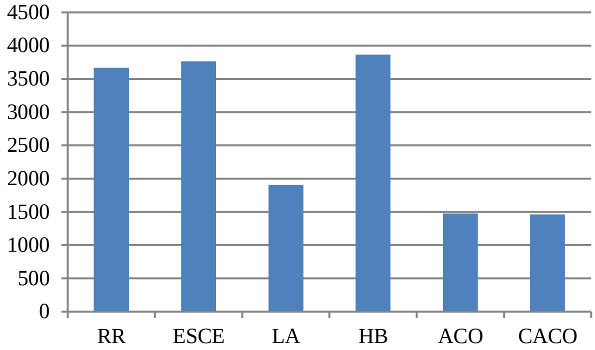


c) S3: Two data centers with 25,50 VMs, located at region 0 and 2 respectively.

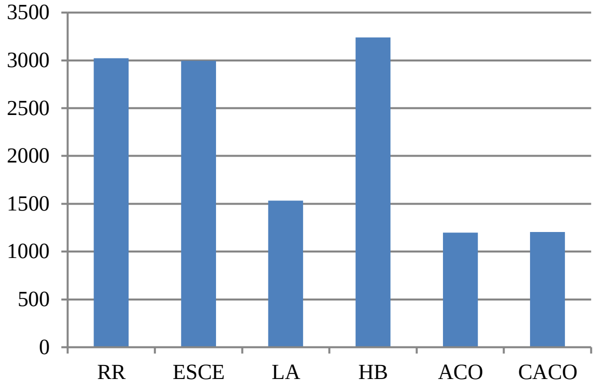


d) S4: Three data centers with 25,30, 50 VMs, located at three different regions 0, 2, and 1 respectively.

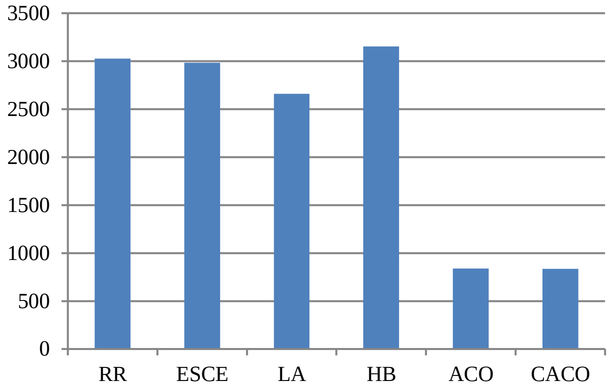
Figure 2. Comparison of processing time (ms) under various scenarios (CDC).



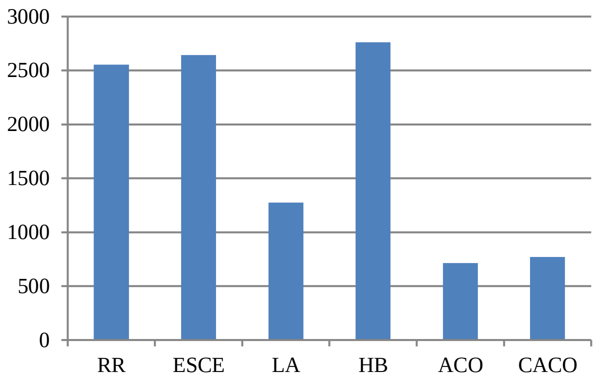
a) S1: One data center with 25 VMs, located at region 0.



b) S2: Two data centers with 25 VMs each, located at region 0 and 2 respectively.



c) S3: Two data centers with 25, 50 VMs, located at region 0 and 2 respectively.

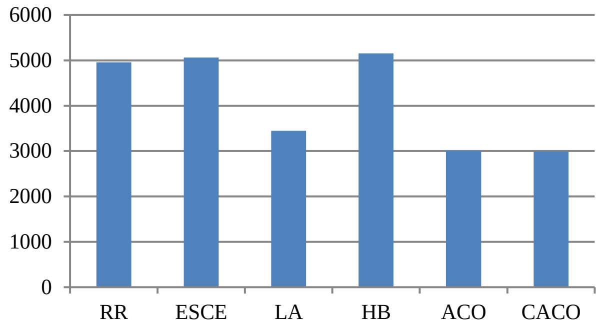


d) S4: Three data centers with 25,30, 50 VMs, located at three different regions 0, 2, and 1 respectively.

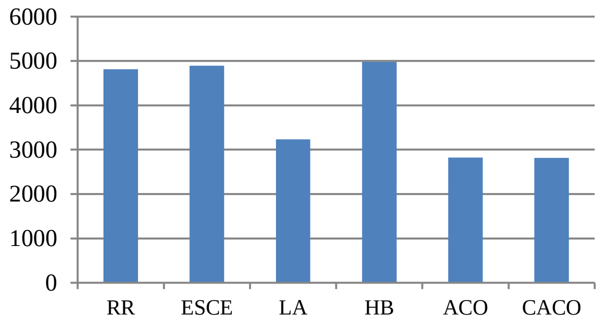
Figure 3. Comparison of processing time (ms) under various scenarios (ORT).

4.2.2. Response Time

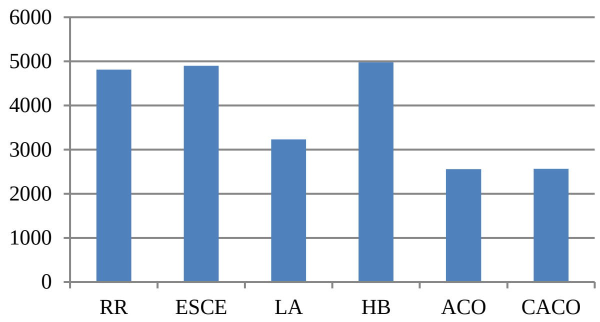
Response time is the time taken by the data center to receive requests from the user base. The response time resulted from the algorithms are represented in milliseconds (ms). The graphical representation of response time under various scenarios for CDC and ORT are shown in Figures 4 and 5 respectively.



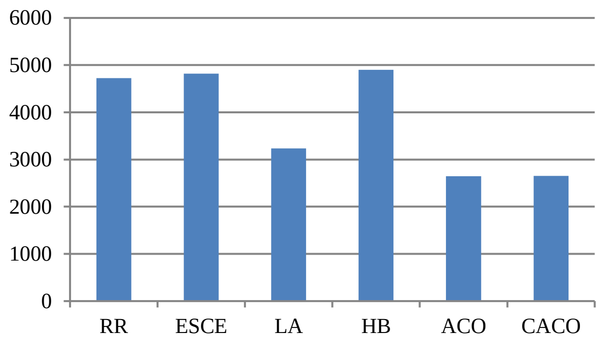
a) S1: One data center with 25 VMs, located at region 0.



b) S2: Two data centers with 25 VMs each, located at region 0 and 2 respectively.

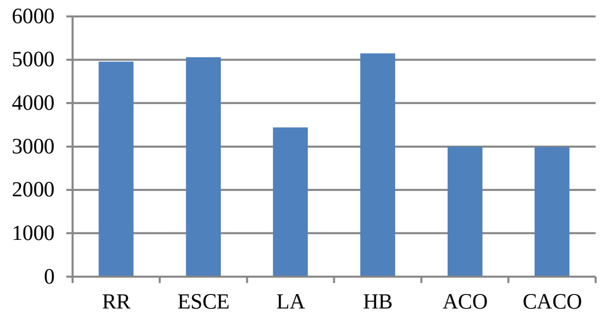


c) S3: Two data centers with 25, 50 VMs, located at region 0 and 2 respectively.

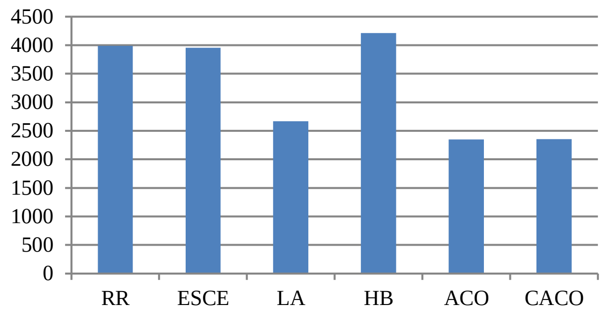


d) S4: Three data centers with 25, 30, 50 VMs, located at three different regions 0, 2, and 1 respectively.

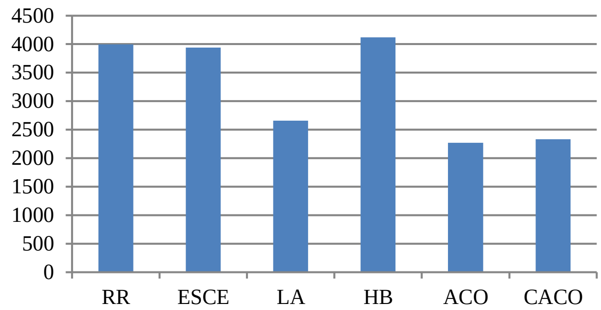
Figure 4. Comparison of Response time (ms) under various scenarios (CDC).



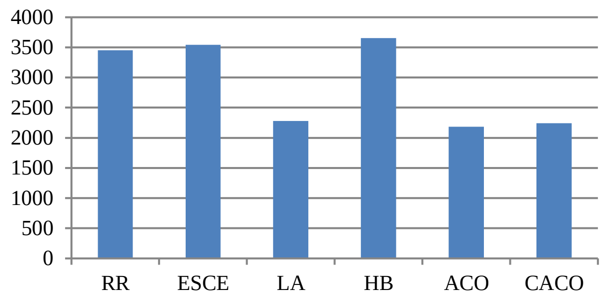
a) S1: One data center with 25 VMs, located at region 0.



b) S2: Two data centers with 25 VMs each, located at region 0 and 2 respectively.



c) S3: Two data centers with 25, 50 VMs, located at region 0 and 2 respectively.



d) S4: Three data centers with 25,30, 50 VMs, located at three different regions 0, 2, and 1 respectively.

Figure 5. Comparison of Response time (ms) under various scenarios (ORT).

From Tables 5 and 6, we can infer that the proposed CACO Algorithm gives better results in terms of processing time and response time respectively when compared to other existing algorithms other than ACO [5]. A quick look-over to the results revealed that, in general, the proposed CACO load balancing algorithm outperforms the other existing algorithms including ACO in scenario1 and scenario 2 with the CDC service broker policy as the overall processing time and the overall response time is comparatively better.

Table 5. Comparative analysis of processing time.

Service broker policy	Load balancing algorithms	Processing Time (ms)											
		S1			S2			S3			S4		
		Min	Max	Average	Min	Max	Average	Min	Max	Average	Min	Max	Average
Closest Data Centre(CDC)	**RR**	0.84	21806.17	3665.93	0.95	21827.41	3747.73	0.95	21827.41	3747.56	0.95	21827.41	3646.30
	ESCE	0.85	22002.57	3761.71	0.95	21998.47	3827.53	0.95	21998.47	3827.38	0.95	21998.47	3737.48
	LA	1.13	18433.90	1907.22	1.13	18442.04	1939.80	1.13	18442.04	1939.82	1.13	18442.04	1894.75
	HB	0.74	22045.58	3865.17	0.94	22045.41	3917.16	0.94	22045.41	3917.24	0.94	22045.41	3832.11
	ACO	0.59	12315.32	1472.31	0.30	12542.06	1551.65	0.46	6462.12	1005.82	0.34	10591.74	1076.16
	CACO	0.44	12041.89	1467.47	0.11	12220.72	1536.39	0.40	6073.58	1013.01	0.34	9604.01	1099.48
Optimize Response Time(ORT)	**RR**	0.74	21921.68	3665.43	0.42	20394.68	3021.56	0.32	20590.02	3025.23	0.31	21863.07	2555.43
	ESCE	0.74	22052.77	3761.66	0.57	20583.83	2995.13	0.43	20583.83	2982.01	0.56	20549.64	2642.84
	LA	0.87	18622.17	1907.24	1.18	17404.51	1533.97	1.18	26519.36	2659.75	0.56	17255.84	1274.77
	HB	0.63	22084.75	3862.56	0.47	22060.21	3238.77	0.36	21335.74	3153.68	0.61	20791.62	2762.32
	ACO	0.61	13753.35	1478.39	0.36	11961.45	1199.31	0.30	6534.64	838.33	0.25	9599.71	713.79
	CACO	0.65	11757.60	1464.39	0.42	10886.67	1204.37	0.32	6268.55	834.18	0.12	9143.17	770.70

Table 6. Comparative analysis of response time.

Service broker policy	Load balancing algorithms	Response Time (ms)											
		S1			S2			S3			S4		
		Min	Max	Average	Min	Max	Average	Min	Max	Average	Min	Max	Average
Closest Data Centre(CDC)	**RR**	187.41	28802.40	4958.24	182.76	28341.49	4812.20	182.76	28341.49	4812.08	182.76	28341.49	4723.75
	ESCE	187.41	29078.84	5058.76	182.76	28863.58	4897.30	182.76	28863.58	4897.09	182.76	28863.58	4819.42
	LA	148.86	28547.56	3443.52	139.36	28351.87	3231.50	139.36	28351.87	3231.53	139.36	28351.87	3235.45
	HB	190.03	29126.29	5151.42	182.76	28974.32	4978.48	182.76	28974.32	4978.48	182.76	28974.32	4902.15
	ACO	152.56	18386.79	2999.04	150.35	18613.98	2821.73	139.28	19543.43	2558.67	144.56	16244.45	2648.50
	CACO	145.60	18446.03	2993.72	144.46	18412.41	2816.04	143.33	19485.51	2562.36	140.71	15908.06	2653.38
Optimize Response Time(ORT)	**RR**	181.46	28657.20	4957.53	102.62	26672.75	3989.20	121.42	26672.75	3995.93	100.96	28444.23	3451.17
	ESCE	181.46	29597.80	5059.27	109.88	27152.84	3954.76	14.96	27152.84	3938.21	88.33	27170.47	3542.93
	LA	141.28	28571.93	3444.01	99.60	26512.16	2668.63	100.23	26519.36	2659.75	103.06	26514.46	2280.09
	HB	181.46	29505.49	5149.31	107.57	28813.53	4214.67	114.43	27605.41	4123.56	101.76	27061.56	3656.81
	ACO	150.30	19664.18	2999.79	105.52	17484.57	2351.26	107.71	19019.52	2272.13	104.87	15564.01	2183.32
	CACO	152.27	18390.22	2992.78	95.62	17264.21	2355.61	104.02	19241.94	2331.38	107.04	15749.67	2241.60

4.2.3. Total Cost

One of the most important parameters of cloud computing is cost. The total cost includes virtual machine migration cost and data transfer cost. During the simulation, the algorithms such as Round Robin, Equally Spread Current Execution Load, Location Aware, and Honeybee give the same results for Total cost as the cost is not included as a factor in these algorithms. The total cost of the algorithms is represented in dollars ($) as shown in Table 7. Hence, these algorithms are altogether referred to as "Others" in Tables 7, 8, 9, and 10. The graphical representation of Total Cost is mentioned in Figure 6.

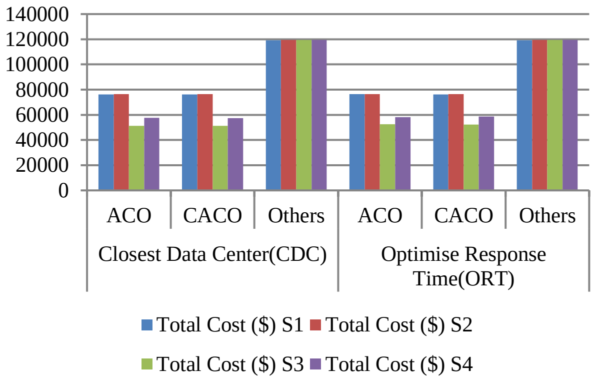


Figure 6. Comparison of total cost ($) under various scenarios.

Table 7. Comparative analysis of total cost.

Service broker policy	Load balancing algorithms	Total Cost ($)			
		S1	S2	S3	S4
Closest Data Centre (CDC)	**ACO**	76303.30	76507.53	51254.80	57545.02
	CACO	76288.00	76436.58	51318.66	57489.16
	Others	119274.92	119370.97	119370.97	119467.02
Optimise Response Time (ORT)	**ACO**	76362.44	76424.50	52615.82	58213.03
	CACO	76281.93	76475.46	52399.65	58583.57
	Others	119274.92	119370.97	119370.97	119467.02

Table 8. Comparative analysis of power consumption.

Service broker policy	Load balancing algorithms	Power (kW)			
		S1	S2	S3	S4
Closest Data Centre (CDC)	**ACO**	10597.6	10626.04	7118.72	7992.36
	CACO	10595.5	10616.19	7127.59	7984.61
	Others	16565.9	16579.30	16579.30	16592.64
Optimise Response Time (ORT)	**ACO**	10605.8	10614.51	7307.75	8085.14
	CACO	10594.7	10621.59	7277.73	8136.61
	Others	16565.9	16579.30	16579.30	16592.64

Table 9. Comparative analysis of energy consumption.

Service broker policy	Load balancing algorithms	Energy(kWh)			
		S1	S2	S3	S4
Closest Data Centre (CDC)	ACO	635860.8	637562.4	427123.2	479541.6
	CACO	635733	636971.4	427655.4	479076.6
	Others	993957.6	994758	994758	995558.4
Optimise Response Time (ORT)	ACO	636353.4	636870.6	438465	485108.4
	CACO	635682.6	637295.4	436663.8	488196.6
	Others	993957.6	994758	994758	995558.4

Table 10. Comparative analysis of carbon footprint.

Service broker policy	Load balancing algorithms	Carbon Footprint(tons)			
		S1	S2	S3	S4
Closest Data Centre (CDC)	ACO	457.8198	459.0449	307.5287	345.27
	CACO	457.7278	458.6194	307.9119	344.9352
	Others	715.6495	716.2258	716.2258	716.802
Optimise Response Time (ORT)	ACO	458.1744	458.5468	315.6948	349.278
	CACO	457.6915	458.8527	314.3979	351.5016
	Others	715.6495	716.2258	716.2258	716.802

4.2.4. Power and Energy Consumption

Power is the rate at which the user requests are satisfied by the data center. Thus, power consumption includes the total power consumed by all the data centers under various scenarios for different algorithms mentioned in Table 8. Power is represented in kilowatt (kW). The graphical representation of Power consumption is mentioned in Figure 7.

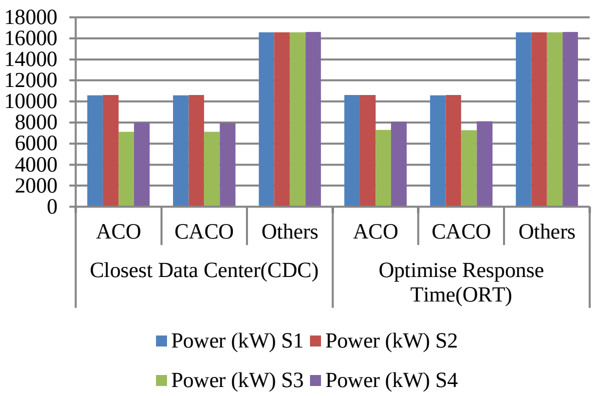


Figure 7. Comparison of Power (kW) under various scenarios.

Data center consumes a large amount of energy because of its high-performance components. Energy is one of the beneficiary factors in the management of data centers in the cloud [22]. Energy is represented in Kilowatt-hour (kWh) for the chosen scenarios represented in Table 9. The graphical representation of energy consumption is mentioned in Figure 8.

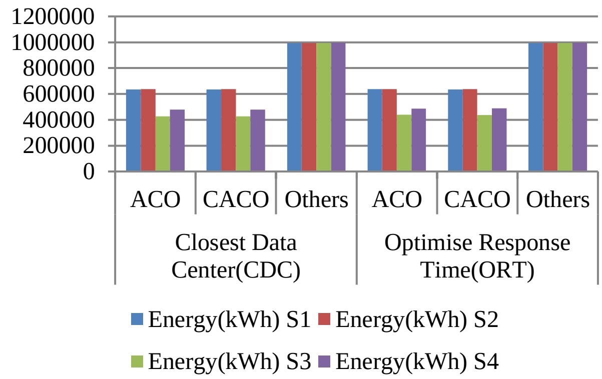


Figure 8. Comparison of energy (kWh) under various scenarios.

4.2.5. Carbon Footprint Reduction

Data center carbon footprint is the amount of carbon released into the atmosphere. By considering the social welfare into account, the CACO model has aimed at reducing the carbon footprint in the cloud environment. Reduction of power usage greatly contributes in reducing the carbon footprint [9]. It is assumed that 1000kWh of power consumption emits 0.72 tons of CO2 [13]. The carbon footprint analysis is represented in Table 10. Carbon footprint under various scenarios for CDC and ORT is graphically represented in Figure 9. The proposed Cost-aware ACO based load balancing model has been compared with other load balancing algorithms.

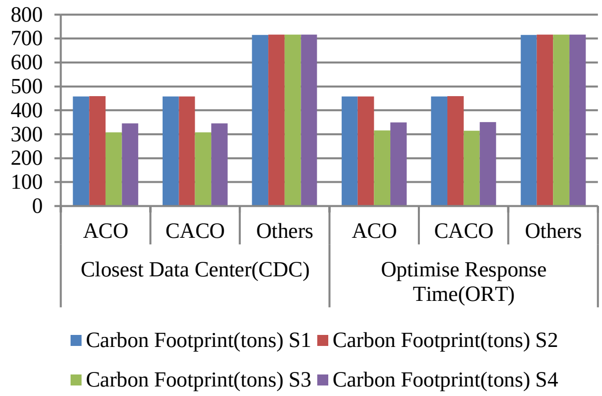


Figure 9. Comparison of carbon footprint (tons) various scenario.

In this paper the statistical t-test analysis of CACO carbon footprint datasets have performed for the service broker policies closed data center and optimised response time. The values of means are392.30 and 395.61 respectively with the standard deviation of 18043.84 and 16394.71.The t-value is -0.06183 and p-value is 0.952. The result is significant since $p > 0.05$.Thus our model gives best result. The main focus of our work is to reduce the carbon footprint in cloud data center. In most of the scenarios, under both service broker policies, our proposed CACO model gives better results in terms of cost, power, energy and carbon footprint. Thus, by managing the carbon footprint efficiently, our model contributes to the betterment and welfare of the environment.

5. Key Challenges

The key challenges faced while analysing and formulating the proposed work are

1. Geographical distribution of nodes across various data centers should consider delay in communication and network, the distance between the resources and users.
2. The algorithm should be designed with multiple nodes to avoid single point of failure.
3. The data load should be evenly distributed across the virtual machines thereby avoiding overloading of virtual machines.
4. Based on the users dynamic requirements the resources should be effectively utilized to minimize the response time in the heterogeneous environment.
5. High scalability, less complexity and efficient storage management helps in formulating efficient cloud system.

6. Conclusions and Future Work

In this paper, we have analyzed the effects of various load balancing algorithms and service broker policies under different scenarios in a large scale cloud environment. The existing algorithms namely, Round Robin, Equally Spread Current Execution Load, Location Aware and Honeybee and Ant Colony algorithms and the service broker policies such as closest data center and optimize response time are taken into consideration in order to analyze the performance of the proposed work. In order to accomplish our work, we have chosen CloudAnalyst simulation tool and FaceBook data set for configuration. The results are tabulated and it clearly revealed that the proposed CACO Algorithm performs better in scenario 1 and scenario 2 with the CDC service broker policy. As power consumption increases, carbon footprint also increases. Therefore, by reducing the power consumption we can reduce the carbon footprint which greatly benefits the environment. The proposed CACO algorithm reduces carbon footprint by 45%. Future work concentrates on finding more procedures to reduce the power consumption and also focus on the power supply for data centers by renewable energy sources [1, 6]. Green cloud data centers powered by renewable energy sources should satisfy the highly dynamic user requirements. The proposed algorithm can be integrated with green cloud data centers thereby contributing towards greener, carbon free cloud environment [7, 20, 25].

References

[1] Butt U., Mehmood M., Shah S., Amin R., Shaukat M., Raza S., Suh D., and Piran M., “A Review of Machine Learning Algorithms for Cloud Computing Security,” *Electronics*, vol. 9, no. 9, pp. 1379, 2020.

[2] Buyya R., “Cloudanalyst: A Cloudsim-Based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments,” *Distributed Computing Project, CSSE Dept., University of Melbourne*, pp.433-659, 2009.

[3] Facebook statistics, www.internetworldstats.com, Last Visited, 2019.

[4] Jain N. and Choudhary S., “Overview of Virtualization in Cloud Computing,” *in Proceeding of Symposium on Colossal Data Analysis and Networking*, Indore, pp. 1-4, 2016.

[5] Jo B., Piran M., Lee D., and Suh D., “Efficient Computation Offloading in Mobile Cloud Computing for Video Streaming Over 5G,” *Computers, Materials and Continua*, vol. 61, no. 1, pp. 439-463, 2019.

[6] Kaluri R., Rajput D., Xin Q., Lakshmanna K., Bhattacharya S., Gadekallu T., and Maddikunta P., “Roughsets-based Approach for Predicting Battery Life in IoT,” *Journal of Intelligent Automation and Soft Computing*, vol. 27, no. 2, pp. 453-469, 2021.

[7] Karamollahi A., Chalechale A., and Ahmadi M., “Energy Consumption Improvement and Cost Saving by Cloud Broker in Cloud Datacenters,” *The International Arab Journal of Information Technology*, vol. 15, no. 3, pp. 405-411, 2018.

[8] Kashyap D. and Viradiya J., “A Survey of Various Load Balancing Algorithms in Cloud Computing,” *International Journal of Scientific and Technology Research*, vol. 3, no. 11, pp. 115-119, 2014.

[9] Khosravi A. and Buyya R., *Sustainable Development: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2018.

[10] Kishor K. and Thapar V., “An Efficient Service Broker Policy for Cloud Computing Environment,” *International Journal of Computer Science Trends and Technology*, vol. 2, no. 4, pp. 104-109, 2014.

[11] Kumar R. and Charu S., “An Importance of Using Virtualization Technology in Cloud Computing,” *Global Journal of Computers and Technology*, vol. 1, no. 2, pp. 56-60, 2015.

[12] Kushwaha M. and Gupta S., “Response Time Reduction and Performance Analysis of Load Balancing Algorithms at Peak Hours in Cloud Computing,” *International Journal of Computer Applications*, vol. 128, no. 17, pp. 26-31, 2015.

[13] Limbani D. and Oza B., “A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation,” *International Journal of Computer Technology and Applications*, vol. 3, no. 3, pp. 1082-1087, 2012.

[14] Meftah A., Youssef A., and Zakariah M., “Effect of Service Broker Policies and Load Balancing Algorithms on The Performance of Large Scale Internet Applications in Cloud Datacenters,” *International Journal of Advanced Computer Science And Applications*, vol. 9, no. 5, pp. 219-227, 2018.

[15] Mell P. and Grance T., “The NIST Definition of Cloud Computing,” Technical Report, National Institute of Standards and Technology, 2011.

[16] Mishra R. and Jaiswal A., “Ant Colony Optimization: A Solution of Load Balancing in Cloud,” *International Journal of Web and Semantic Technology*, vol. 3, no. 2, pp. 33, 2012.

[17] Naqvi S., Javaid N., Butt H., Kamal M., Hamza A., and Kashif M., “Metaheuristic Optimization Technique for Load Balancing in Cloud-Fog Environment Integrated with Smart Grid,” *in Proceeding of International Conference on Network-Based Information Systems*, Bratislava, pp. 700-711, 2018.

[18] Nitika M., Shaveta M., and Raj M., “Comparative Analysis of Load Balancing Algorithms in Cloud Computing,” *International Journal of Advanced Research in Computer Engineering and Technology*, vol. 1, no. 3, pp. 120-124, 2012.

[19] Patel H. and Patel R., “Cloud Analyst: An Insight of Service Broker Policy,” *International Journal of Advanced Research in Computer and Communication Engineering*, vol. 4, no. 1, pp. 122-127, 2015.

[20] Priya S., Bhattacharya S., Maddikunta P., Somayaji S., Lakshmanna K., Kaluri R., Hussien A., and Gadekallu T., “Load Balancing of Energy Cloud Using Wind Driven and Firefly Algorithms in Internet of Everything,” *Journal of Parallel and Distributed Computing*, vol. 142, pp. 16-26, 2020.

[21] Rehman M., Javaid N., Ali M., Saif T., Ashraf M., and Abbasi S., “Threshold Based Load Balancer for Efficient Resource Utilization of Smart Grid Using Cloud Computing,” *in Proceeding of International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, Taichung, pp. 167-179, 2018.

[22] Rong H., Zhang H., Xiao S., Li C., and Hu C., “Optimizing Energy Consumption for Data Centers,” *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 674-691, 2016.

[23] Subalakshmi S. and Malarvizhi N., “Enhanced Hybrid Approach for Load Balancing Algorithms in Cloud Computing,” *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2, no. 2, pp. 136-142, 2017.

[24] Thakur S. and Chaurasia A., “Towards Green Cloud Computing: Impact of Carbon Footprint on Environment,” *in Proceeding of 6th International Conference - Cloud System and Big Data Engineering*, Noida, pp. 209-213, 2016.

[25] Thilagavathi N., Subha R., and Uthariaraj V., “Eco-Aware Load Balancing for Distributed Cloud Data Centers with Renewables,” *in Proceeding of 10th International Conference on Advanced Computing*, Chennai, pp. 229-236, 2018.

[26] Uddin M., Darabidarabkhani Y., Shah A., and Memon J., “Evaluating Power Efficient Algorithms for Efficiency and Carbon Emissions in Cloud Data Centers: A Review,” *Renewable and Sustainable Energy Reviews*, vol. 51, pp. 1553-1563, 2015.

[27] Xing Y. and Zhan, Y., “Virtualization and Cloud Computing,” *in Proceeding of Future Wireless Networks and Information Systems*, *Lecture Notes in Electrical Engineering*, Wuhan, pp. 305-312, 2012.

[28] Zhao W., Peng Y., Xie F., and Dai Z., “Modeling and Simulation of Cloud Computing: A Review,” *in Proceeding of Asia Pacific Cloud Computing Congress*, Shenzhen, pp. 20-24, 2012.

Malini Alagarsamy obtained her PhD in Information and Communication Engineering from Anna University, Chennai. She is currently an assistant professor at Thiagarajar College of Engineering, Madurai, India. She has published several research papers in journals and international/national conferences. Her research interest includes software Engineering, Testing, Mobile Application development, Green Computing, Internet of Things, Block chain and Machine Learning.

Ajitha Sundarji obtained her B.E in Computer Science and Engineering from Thiagarajar College of Engineering, Madurai, India in 2019.She is currently an associate engineer at AstraZeneca GTC, Chennai, India. Her research interest includes Software testing, Scheduling, Resource allocation and Green computing.

Aparna Arunachalapandi obtained her B.E in Computer Science and Engineering from Thiagarajar College of Engineering, Madurai, India in 2019.She is currently a software engineer at Wipro Technologies, Bangalore, India. Her research interest includes Software testing, Scheduling, Resource allocation and Green computing.

Keerthanaa Kalyanasundaram obtained her B.E in Computer Science and Engineering from Thiagarajar College of Engineering, Madurai, India in 2019.She is currently a software developer at HCL Technologies Limited, Elcot sez park, Madurai, India. Her research interest includes Software testing, Scheduling, Resource allocation and Green computing.