Computer Science and Information Systems 21(3):759–780 https://doi.org/10.2298/CSIS230923018P

SSEPC Cloud: Carbon Footprint Aware Power Efficient Virtual Machine Placement in Cloud Milieu

Bivasa Ranjan Parida[1,⋆], **Amiya Kumar Rath**[1,2], **Bibudhendu Pati**[3], **Chhabi Rani Panigrahi**[3], **Hitesh Mohapatra**[4], **Tien-Hsiung Weng**[5,⋆], **and Rajkumar Buyya**[6]

[1] Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, Odisha, India
bivasa.parida@silicon.ac.in
[2] Vice-Chancellor, Biju Patnaik University of Technology, Rourkela, Odisha, India
akrath_cse@vssut.ac.in
[3] Department of Computer Science, Rama Devi Women's University, Bhubaneswar, Odisha, India
patibibudhendu@rdwu.ac.in
panigrahichhabi@gmail.com
[4] School of Computer Engineering, KIIT (Deemed to be) University, Bhubaneswar, Odisha, India
hiteshmahapatra@gmail.com
[5] Department of Computer Science and Information Engineering, Providence University, Taiwan
thweng@gm.pu.edu.tw
[6] Cloud Computing and Distributed Systems (CLOUDS) Lab, School of Computing and Information Systems, University of Melbourne, Australia
rbuyya@unimelb.edu.au

Abstract. The consumption of energy and carbon emission in cloud datacenters are the alarming issues in recent times, while optimizing the average response time and service level agreement (SLA) violations. Handful of researches have been conducted in these domains during virtual machine placement (VMP) in cloud milieu. Moreover it is hard to find researches on VMP considering the cloud regions and the availability zones along with the datacenters, although both of them play significant roles in VMP. Hence, we have worked on a novel approach to propose a hybrid metaheuristic technique combining the salp swarm optimization and emperor penguins colony algorithm, i.e. SSEPC to place the virtual machines in the most suitable regions, availability zones, datacenters, and servers in a cloud environment, while optimizing the mentioned quality of service parameters. Our suggested technique is compared with some of the contemporary hybrid algorithms in this direction like Sine Cosine Algorithm and Salp Swarm Algorithm (SCA-SSA), Genetic Algorithm and Tabu-search Algorithm (GATA), and Order Exchange & Migration algorithm and Ant Colony System algorithm (OEMACS) to test its efficacy. It is found that the proposed SSEPC is consuming 4.4%, 8.2%, and 16.6% less energy and emitting 28.8%, 32.83%, and 37.45% less carbon, whereas reducing the average response time by 11.43%, 18.57%, and 26% as compared to its counterparts GATA, OEMACS, and SCA-SSA respectively. In case of SLA violations, SSEPC has shown its effectiveness by lessening the value of this parameter by 0.4%, 1.2%, and 2.8% as compared to SCA-SSA, GATA, and OEMACS respectively.

Keywords: Virtual Machine Placement, Energy Consumption, Carbon Emission, Salp Swarm Optimization, Emperor Penguins Colony Algorithm.

⋆ Corresponding authors

1. Introduction

Cloud computing has proved itself as an indispensable part of technologies in recent times being a utility as pay-per-use model to deliver the resources to the cloud users from the datacenters virtually [1]. The significant cost of expense in the datacenters is the cost of energy consumption [2]. In 2020, the datacenters globally used 1% of world electricity [3]. If the remedial measures have not taken and the technological trends remain unchanged for the coming decade, then between 2016 and 2030, the energy consumption by the datacenters may exceed with 12% approx. [4]. Some countries also enforce tax on carbon footprint for the environmental sustainability [5]. Carbon footprint can be reduced by implementing more and more renewable energy in cloud datacenters [6]. Due to the erratic availability of renewable energy round the clock, the hybrid model of power supply including the brown energy obtained from the fossil fuels along with the green energy from the renewable sources are used in the cloud datacenters [6]. Hence, the consumption of energy and carbon emission in cloud datacenters are the genuine issues to deal with, while optimizing the average response time and service level agreement (SLA) violations for better quality of service (QoS). The strategy of virtual machine placement (VMP) directly influences the consumption of energy and carbon footprints of a datacenter [7] along with the mentioned QoS parameters.

1.1. Motivation and Contributions

Although a handful of researches have been conducted to minimize the energy consumption and carbon emission in cloud datacenters, efficient methodologies are still scarce. Energy consumption by the networks and the cooling system in a datacenter are ignored in most of the research works. Carbon footprint estimation and minimization is a matter of concern for maintaining the environmental sustainability. Addressing these issues through VMP in the cloud datacenters while maintaining the average response time along with SLA violations, are the motivating factors behind this research. Although availability zones are purely logical entities in the cloud regions [8], both of them play significant roles in mapping the requests onto the physical servers in the datacenters. The user account in service provider's domain determines the list of cloud regions accessible to that user by default. Some other regions may be enabled by the user and some others may be restricted depending on global policies. If the region selection is not done by the user, then the process is automated by the service provider through certain searching mechanisms to allot a default region [9]. Multiple availability zones as the isolated locations exist in each region. Expansion of an availability zone may be constrained due to the increasing number of zones in a region. So, for the new users, the constrained availability zones may be removed from the list. Two different users may access different set of availability zones while availing services from the same cloud region. If user doesn't insist on an availability zone, then the service provider selects one zone for the user. It is always recommended for the initial request of a user to continue with the default availability zone selected by the service provider through certain searching mechanisms [9]. It is hard to find works conducted so far for VMP considering cloud regions and availability zones along with datacenters and the servers to the best of our knowledge. Hence, we are motivated to contribute a little to this vast field of emerging aspect of research in cloud milieu. The following points are the key contributions of this work.

- This proposed methodology introduces a hybrid nature inspired optimization technique to solve the problem of VMP in the cloud milieu in two phases. In the first phase, the Salp Swarm Algorithm (SSA) [10] is used to explore geographically in the global search space to find out the suitable cloud region and further converge to the most appropriate availability zone with the aim to make the best use of green energy. Then it exploits to find out the best datacenter possible for VMP with the availability of maximum green energy. The Emperor Penguins Colony (EPC) algorithm [11] enriched with the potential of local search has been implemented in the second phase, to trace the most suitable server in the datacenter for placing the virtual machines (VMs). Both the algorithms are hybridized in the proposed algorithm Salp Swarm based Emperor Penguins Colony (SSEPC) for overcoming the limitations of each other to establish a robust methodology.
- SSEPC places the VMs appropriately to minimize energy consumption and carbon emission of a datacenter with respect to average response time and SLA violations.
- Proposed VMP model considers the energy consumption due to servers, networks and the cooling system of the datacenters simultaneously to make the approach more realistic in the cloud environment.
- Unlike other existing VMP proposals, this model portrays both initial as well as the runtime VMP in the cloud milieu considering the priority of user requests.

The rest of the paper is organized as follows. Section 2 reviews the literature of VMP, consumption of energy, and carbon emission in cloud environment along with the QoS parameters. The system model along with the mathematical formulations are elucidated in Section 3. The proposed algorithm is explored in the Section 4 and Section 5 deliberates the simulation environment, as well as analyses the experiments conducted in heterogeneous cloud environment and the obtained results. Finally, the conclusive remarks along with the future directions in the research in this domain are illuminated in the Section 6.

2. Related Work

The process of allocating a VM to a server in the datacenter to avail the required resources to host the application requested by the user is called as VMP [12]. From the existing works related to VMP in the cloud milieu, it has been observed that although works have been conducted on inter datacenter or multi datacenter VMP [2], [6], [7], [15], [16] considering the assignment of VMs to the appropriate datacenters followed by mapping onto the suitable servers, we didn't come across any VMP proposal considering mapping of VMs onto the compatible cloud regions and the availability zones. Technically a cloud region contains multiple availability zones, an availability zone may be a collection of single or many datacenters, and a datacenter physically holds huge number of servers [8].

The energy consumption by a datacenter may be broadly categorized into two types; i.e. (i) The consumption of energy by the IT equipment like servers and networks, (ii) The usage of energy by the non-IT equipment like cooling system [2]. Hence, none of the components can be ignored while designing an energy conservation model [2]. Minimization of carbon emission in the datacenters is becoming a necessity now-a-days, as carbon taxes are imposed in some countries and in others carbon footprints are taken seriously for the environmental sustainability [5]. The response time [6], [7], [17] and SLA violations [6], [7], [14], [15], [16], [17], [19] play vital role in cloud milieu to meet the quality of experience (QoE) of the users during VMP in cloud datacenters.

Table 1 compares the proposed work SSEPC with the other relevant proposals of recent times in terms of different performance metrics like datacenter energy consumption based on server, networks, and cooling equipment, carbon emission, response time, SLA violations, and use of renewable energy sources for environmental sustainability. Hybridizing multiple methodologies helps in overcoming the limitations of one another [24]. The hybridization is one of the most suitable strategies accepted widely for promoting diversity in the search space to achieve global optimum [24]. Hence, we have hybridized two recently proposed metaheuristic techniques SSA (2017) [10] and EPC (2019) [11] to search the global space for VMP. Table 2 analyses some existing hybrid algorithms for VMP in cloud domain for comparison with the proposed SSEPC methodology.

Table 1. Analogy of related works with proposed work in terms of various metrics

Proposed By	**VMP Implemented**	**Hybrid Approach**	**Server Based Energy Consumption**	**Network Based Energy Consumption**	**Cooling Based Energy Consumption**	**Carbon Emission**	**Response Time**	**SLA Violation**	**Use of Green Energy**
Feng et al. [2]	✓	✓	✓	✓	✓				
Xu et al. [6]	✓		✓		✓	✓	✓	✓	✓
Abbasi-khazaei et al. [7]	✓		✓			✓	✓	✓	✓
Liu et al. [12]	✓	✓	✓						
Zhao et al. [13]	✓	✓	✓						
Samriya et al. [14]	✓		✓					✓	
Gharehpasha et al. [15]	✓	✓	✓					✓	
Zhao et al. [16]	✓	✓	✓			✓		✓	✓
Xu et al. [17]	✓		✓				✓	✓	
Tang et al. [18]	✓	✓	✓	✓					
Yadav et al. [19]	✓		✓	✓				✓	
Khodayarseresht et al. [20]	✓		✓			✓			
Belabed et al. [21]	✓			✓					
Justafort et al. [22]	✓	✓	✓	✓	✓	✓			
Anusooya et al. [23]		✓	✓			✓			
Proposed Algorithm	✓	✓	✓	✓	✓	✓	✓	✓	✓

3. System Model

This section elaborates the system components and the system parameters with mathematical elucidations for designing the proposed model for VMP to satisfy the QoS metrics. Figure 1 portrays the system model of the proposed methodology.

Table 2. Existing hybrid algorithms of VMP considered for comparison

Authors	Name of the Hybrid Algorithm	Algorithms involved in Hybridization	Purpose of Hybridization
Liu et al. [12]	OEMACS	Order Exchange and Migration algorithm and Ant Colony System algorithm	OEM swaps VMs between servers locally, whereas ACS allocates VMs in minimum number of servers through global search
Zhao et al. [13]	GATA	Genetic Algorithm and Tabu-search Algorithm	The algorithm of tabu-search works like a mutation operator of genetic algorithm for improving its local search ability
Gharehpasha et al. [15]	SCA-SSA	Sine Cosine Algorithm and Salp Swarm Algorithm	SCA explores and exploits the space of searching for finding the optimal solution, whereas SSA manages leader and followers in the population to reach at the optimal solution

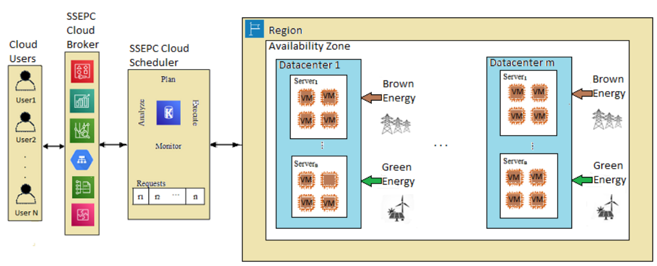


Fig. 1. SSEPC based Cloud System Model for Virtual Machine Placement

3.1. System Components

The cloud service providers, cloud brokers, and cloud users are the entities of a cloud system. The infrastructure of the cloud is built around physically separated regions in different geographic locations [8]. Regions contain multiple availability zones to ensure resiliency [25]. Each availability zone consists of one or more datacenters [8]. A datacenter contains $50,000$ to $80,000$ physical servers [26]. Cloud broker works like an interface for the cloud users to communicate with the service provider. Here, the SSEPC cloud broker plays the major role of routing the user requests to the most suitable resources for the minimization of the consumption of energy and emission of the carbon footprints while maximizing the renewable energy utilization, in such a way that the average response time and SLA violations remain as minimum as possible. The process of scheduling the requests to the servers of the different datacenters belongs to the MAPE-K [27] loop in Figure 1. This loop consists of different phases, such as Monitor, Analyse, Plan, and Execute, managed by the Knowledge module. Lower the number of rejection of the requests to create VMs, higher is the QoE of the cloud users.

The set of u cloud regions is represented as a set $CR = \{cr_1, cr_2, cr_3, \ldots, cr_u\}$, where each $cr_x \in CR$, such that $1 \leq x \leq u$. A region contains v availability zones as a set $AZ = \{az_1, az_2, az_3, \dots, az_v\}$, where each $az_y \in AZ$, such that $1 \le y \le v$. The set of m datacenters in an availability zone is represented as a set $D = \{d_1, d_2, d_3, \dots, d_m\}$, where each $d_i \in D$, such that $1 \le i \le m$. Each datacenter has n number of physical servers, i.e., $S = \{s_1, s_2, s_3, \dots, s_n\}$, where $s_j \in S$, such that $1 \le j \le n$. Each server possesses p number of VMs portrayed as set $VM = \{vm_1, vm_2, vm_3, \dots, vm_p\}$, where $vm_k \in VM$, such that $1 \le k \le p$. R is the set of requests, i.e., $R = \{r_1, r_2, r_3, \dots, r_q\}$, where $r_l \in R$, such that $1 \le l \le q$. Hence, the relationship between the cloud regions, availability zones, datacenters, servers, and the VMs can be defined as follows.

$$\{\{\{\{\{vm_1, vm_2, \dots, vm_p\}, s_2, \dots, s_n\}, d_2, \dots, d_m\}, az_2, \dots, az_v\}, cr_2, \dots, cr_u\}$$

3.2. System Parameters

This subsection discusses the system parameters mentioned as follows.

3.2.1 Energy Consumption of Datacenters The components mainly responsible for the consumption of energy in a datacenter are the servers, networks, and cooling system [2]. Hence, energy consumed by all the datacenters E_D can be illustrated as in Eqn. (1).

$$E_D = \sum_{i=1}^{m} E_{d_i} = \sum_{i=1}^{m} (E_{i,Server} + E_{i,Network} + E_{i,Cooling}) \tag{1}$$

Here, the energy consumed in total by a datacenter is E_{d_i}, where $E_{i,Server}$, $E_{i,Network}$, and $E_{i,Cooling}$ are the amount of energy consumption by the servers, networks, and the cooling system of a datacenter respectively.

3.2.1.1 Server Energy Consumption Model The energy consumed by all the servers of a datacenter d_i is denoted as $E_{i,Server}$, which is given [28] as in Eqn. (2).

$$E_{i,Server} = \sum_{j=1}^{n} E_{i,s_j} = \sum_{j=1}^{n} (E_{j,Idle} + (E_{j,Max} - E_{j,Idle}) \times u_{s_j}) \tag{2}$$

Here, consumption of energy by a server s_j in a datacenter d_i can be represented as E_{i,s_j}, where $E_{j,Idle}$ and $E_{j,Max}$ are the energy consumed by the server s_j at idle time and peak time respectively, u_{s_j} is the utilization of j^{th} server.

3.2.1.2 Network Energy Consumption Model Let the communication graph of VMs be $G_C(VM, L_C)$. Here, VM represents the set of VMs and L_C stands for the set of edges or links between the VMs for communication among themselves, such that $v_1, v_2 \in VM$ and edge$(v_1, v_2) \in L_C$. The bandwidth required for this communication is $bw(v_1, v_2)$. Let the datacenter network topology graph be $G_T(S, L_T)$. Here, S represents the set of servers and L_T stands for the set of edges or links between servers, such that $s_1, s_2 \in S$ and edge$(s_1, s_2) \in L_T$, only if s_1 links to s_2 in network topology. The bandwidth associated with each link can be represented as $bw(s_1, s_2)$. Let f_v be the function to determine the result of VMP, i.e., $s_1 = f_v(v_1)$, if and only if the VM v_1 is placed on the server s_1. Let the length of shortest path between v_1 and v_2 be $P(v_1, v_2)$. The consumption of energy by the network of a datacenter d_i is denoted as $E_{i,Network}$, which can be formulated [2] as in Eqn. (3).

$$E_{i,Network} = \sum_{(v_1,v_2)\in L_C} bw(v_1, v_2) \times P(v_1, v_2) \tag{3}$$

3.2.1.3 Cooling Energy Consumption Model The cooling system of a datacenter is called as Computer Room Air Conditioning (CRAC) [2]. Let the maximum temperature of air flow supplied by CRAC is $T_{Sup(CRAC)}$. It is inversely proportional to the energy consumption of cooling $E_{i,Cooling}$. Wang et al. [29] found that up to $4.3 - 9.8\%$ energy can be conserved per 1°C rise in $T_{Sup(CRAC)}$. The cooling system energy consumption $E_{i,Cooling}$ can be calculated based on the proposed model of Tang et al. [30] and is given in Eqn. (4).

$$E_{i,Cooling} = \frac{E_{i,Server} + E_{i,Network}}{0.0068 \times (T_{Sup(CRAC)})^2 + 0.008 \times T_{Sup(CRAC)} + 0.458} \tag{4}$$

3.2.2 Renewable Energy Availability Cloud service providers are swiftly moving towards the renewable energy sources to derive electricity for the datacenters to reduce the utilization of brown energy due to its high energy cost, carbon footprint cost and alarming negative impact on environment. Although most of the renewable energy sources are intermittent in nature, this proposed technique considers the use of renewable or green energy (E^G) in cloud datacenters with highest priority as compared to the brown energy (E^B). Datacenters will utilize the renewable energy first as long as it is available before using the brown energy.

3.2.3 Carbon Emission of Datacenters The approach for minimizing the carbon emission in datacenters using SSEPC is based on the carbon intensity value instead of relying on weather data for VMP, as weather data may be an inaccurate metric. According to National Grid Electricity System Operator, carbon intensity is the number of grams of CO2 emitted per unit of electricity at one kilowatt hour [31]. Carbon intensity value is based on the type of brown energy, where this value is zero for green energy [7]. If the energy required by the datacenter E_{d_i} is greater than the green energy $E^G_{d_i}$ available in that datacenter, then the brown energy $E^B_{d_i}$ usage will have a positive value. On the contrary, if the available green energy is more than the energy requirement of the datacenter, then the brown energy usage value will be zero. Hence, the total carbon emission [6] from all the datacenters C_D is defined as in Eqn. (5).

$$C_D = \sum_{i=1}^{m} C_{d_i} = \sum_{i=1}^{m} (E^B_{d_i} \times I^C_{d_i}) = \sum_{i=1}^{m} (max(E_{d_i} - E^G_{d_i}, 0) \times I^C_{d_i}) \tag{5}$$

Where C_{d_i} is the total carbon emission of the datacenter d_i and $I^C_{d_i}$ is the carbon intensity of datacenter d_i.

3.2.4 Response Time For the better QoE of the cloud users, response time plays a vital role along with other parameters. To make the proposed model realistic one, the response time of the requests is considered along with other parameters. Let the response time for request r_l allocated to d_i is T_{l,d_i}. Due to insufficient green energy, if the request is forwarded to another datacenter $d_i^{'}$, then the response time will be $T_{d_i,d_i'}$. Hence, the response time of r_l is $T_{r_l} = T_{l,d_i} + T_{d_i,d_i'}$. The average response time of all requests T_D can be calculated as in Eqn. (6).

$$T_D = \frac{1}{|R|} \sum_{r_l \in R} T_{r_l} \tag{6}$$

3.2.5 Total Cost of Execution The purpose of designing the proposed model is to minimize the consumption of energy and emission of carbon footprints in a cloud datacenter, while optimizing the average response time. Hence, the total cost of execution of the requests in cloud datacenters C_T can be composed of the total energy consumed by the datacenters E_D, the total carbon emission from the datacenters C_D, and the average response time of all requests in the system T_D and is given in Eqn. (7).

$$C_T = E_D + C_D + T_D \tag{7}$$

3.2.6 Capacity of a Server The capacity of a server s_j in a datacenter d_i can be represented as Cap_{i,s_j} in Eqn. (8) [32].

$$Cap_{i,s_j} = \sum_{k=1}^{p} Cap_{j,vm_k} = \sum_{k=1}^{p} (num(PEs)_{vm_k} \times MIPS(PEs)_{vm_k} + BW_{vm_k}) \tag{8}$$

Where the capacity of a VM vm_k in a server s_j is Cap_{j,vm_k}, $num(PEs)_{vm_k}$ is denoted as the count of processing elements in k^{th} VM, $MIPS(PEs)_{vm_k}$ stands for the million instructions per second of all the processing elements of k^{th} VM, and BW_{vm_k} represents the bandwidth of k^{th} VM.

3.2.7 SLA Violations SLA violations in datacenters arise due to several reasons. Overuse of a server or the resources requested by a VM not allocated by the server etc. may be the reasons of SLA violation related to our current context. Minimization of SLA violation is required for better QoE of the users. The overall SLA violation value SLA_{vv} is calculated as in Eqn. (9) [14].

$$SLA_{vv} = \frac{MIPS_{TR} - MIPS_{TA}}{MIPS_{TR}} \tag{9}$$

Here, $MIPS_{TR}$ is the total million instructions per second requested and $MIPS_{TA}$ is the total million instructions per second assigned to VMs on the basis of user demands.

3.2.8 Total Optimum Value On regular intervals, the proposed algorithm calculates the values of energy consumption, SLA violations, and capacity of the servers of the datacenter for the VMP. The value of energy consumption of a server s_j in a datacenter d_i is calculated in Eq.(2) as E_{i,s_j} and SLA violation value SLA_{vv} is calculated as in Eqn. (9).

$$O_T = E_{i,s_j} + SLA_{vv} \tag{10}$$

Here, O_T is the total optimum value, which is the combination of energy consumption value of a server E_{i,s_j} and the SLA violation value SLA_{vv}.

4. Salp Swarm optimization based Emperor Penguins Colony (SSEPC) Algorithm for Virtual Machine Placement

This section elaborates the proposed SSEPC algorithm. It is of multi-stage collaborative hybrid type [24] comprising of two phases, where SSA performs global search in Phase-I of SSEPC and the EPC algorithm does the local search in SSEPC Phase-II. Both SSA and EPC are swarm inspired stochastic collective algorithms [10].

4.1. SSEPC Phase-I based on the Salp Swarm Algorithm

In SSEPC Phase-I, the VMs are considered as the salps in the population foraging the food sources, where the computing resources in the datacenters mimic the foods.The salps compete among themselves to discover the best source of food in the ocean. Similarly, the VMs also compete during VMP for availing the most suitable regions, availability zones, and the datacenters with respect to maximizing the green energy utilization with the objective of minimization of consumption of energy, emission of carbon footprints, and average response time.

According to Eqn. (11) [10], the position of the leader salp in the salp chain foraging the food sources is updated.

$$X_j^1 = \begin{cases} F_j + r_1\big((UB_j - LB_j)r_2 + LB_j\big), & r_3 \geq 0, \\ F_j - r_1\big((UB_j - LB_j)r_2 + LB_j\big), & r_3 < 0. \end{cases} \tag{11}$$

Where the leader's position at j^{th} dimension is represented as X_j^1 and F_j portrays the location of destined resource in the same dimension j. The upper bound as well as the lower bound of j^{th} dimension are portrayed as UB_j and LB_j respectively. r_2 and r_3 are the random numbers generated evenly between 0 and 1. The r_1 is also a random value, which acts as the significant factor for trading off between exploration and exploitation. The Eqn. (12) is used to calculate r_1 [10].

$$r_1 = 2e^{\left(-\frac{4t}{T_{max}}\right)^2} \tag{12}$$

Where T_{max} stands for the number of iterations at max and the iteration at an instant is mimicked by t. With the increasing values of the number of iterations, the parameter r_1 goes down. In this way it becomes capable of maintaining the uniformity among the exploration and the exploitation.

The Newton's law of motion inspires the Eqn. (13) to update the position of the follower VMs [10].

$$X_j^i = \frac{(X_j^i + X_j^{i-1})}{2} \tag{13}$$

Where i^{th} follower's position in j^{th} dimension is X_j^i, such that $i \geq 2$. The SSA is meant for providing solutions to the optimization problems of continuous nature. But VMP in cloud computing is of binary characteristic, as the VMs migrate in a restricted fashion confined to 0 or 1. Hence, the generated solutions are required to be transformed into discrete values. Transfer function (TF) is a convenient way for the transformation of continuous solutions into the discrete ones [33]. In our work, we have used sigmoid transfer function for this purpose in Eqn. (14).

$$S(X_j^{i+1}) = sig(X_j^{i+1}) = \frac{1}{1 + e^{-X_j^{i+1}}} \tag{14}$$

Where

$$X_j^{i+1} = \begin{cases} 1, & if S(X_j^{i+1}) > \text{ran.nextInt(2)}, \\ 0, & otherwise. \end{cases}$$

Computational overhead for searching the global space significantly decreases due to the implementation of salp swarm optimization in the first phase, as the requests from the same user follow the path of the first request without wasting time for further searching. The objective is to search the appropriate datacenter in an availability zone of a particular region of the cloud milieu for VMP by minimizing the total cost as per the Eqn. (7) in the first phase. Hence, the multi objective fitness function of the Phase-I of proposed SSEPC, F_{val_1} is as in Eqn. (15).

$$F_{val_1} = min.(C_T) \tag{15}$$

4.2. SSEPC Phase-II based on the Emperor Penguins Colony Algorithm

The SSEPC methodology implements EPC algorithm in its second phase for VMP to search the most suitable server in a datacenter locally. Here, the servers mimic the herd of the penguins and the VMs imitate the penguins. List of VM requests (penguins) approaching the datacenter is collected. They search for the appropriate server (huddle). A penguin intends to be warmed goes to a nearby huddle randomly. Likewise a VM is allotted to a server in the datacenter arbitrarily. If the huddle has maximum number of penguins, then the new penguin arrived recently will remain at the periphery and can't be warmed enough. So this penguin will again move to another huddle having less than maximum number of penguins to be warmed appropriately. But if a huddle contains very less number of penguins, then the penguins loss a lot of energy and enough heat can't be generated in that huddle. So, those penguins prefer to move to an existing huddle with sufficient capacity to be accommodated, by dissolving their huddle. Similarly, a new VM migrates to an appropriate server accommodating less than maximum VMs, as it can't be allocated with resources by a server already reached to its maximum capacity. Likewise, if a server is allocated with very less number of VMs, it lets the VMs to migrate to another server and goes to sleep mode to conserve energy. This is the process of VM migration in running VMP.

The capacity of a server s_j at an instance of time should be within a threshold value depending on the physical resource availability of the server to allow a new VM to be allocated to that server. If Cap_{i,s_j} will go beyond the upper bound threshold value (Th_{UB}) after adding the new VM, then instead of allocating resources to that VM, the server will forward it to the VM migration list. Similarly, if Cap_{i,s_j} will fall below the lower bound threshold value (Th_{LB}) in a server, then the allocated VMs will be added to the VM migration list and the server will go to sleep mode to conserve energy. According to Xu et al. [6], sleep mode is for 0% server utilization, idle mode is when server utilization remains less than or equal to 50% and 100% server utilization denotes the maximum. Hence, the servers being under utilized are sent to sleep mode after migrating their allocated tasks help in conserving energy and reducing computational cost.

On the basis of priority, the VMs in the VM migration list will be allocated to other suitable servers. The priority has been calculated by using priority-aware VM allocation (PAVA) algorithm [34] and set to a binary value, i.e. either critical or not. The objective is to minimize O_T, while satisfying the server capacity. A multi objective fitness function F_{val_2} can be considered for VMP inside a datacenter using the Phase-II of proposed SSEPC algorithm and is given in Eqn. (16).

$$F_{val_2} = \underset{s.t.\ Th_{LB} \leq Cap_{i,s_j} \leq Th_{UB}}{min.} O_T \tag{16}$$

4.3. Proposed SSEPC Algorithm

The proposed SSEPC for VMP is presented in the Algorithm 1. From a user, the first request to search an appropriate cloud region, followed by the availability zone and then the most suitable datacenter, is treated as the leader salp. Then the other requests from the same user behave like follower salps on the basis of updated data received from the leader, as proposed in the Phase-I of SSEPC. Algorithm 2, Algorithm 3, and Algorithm 4 are the various modules of Phase-I to search the cloud regions, availability zones, and the datacenters respectively. Algorithm 5 represents the VM migration during running VMP among the servers of a datacenter based on EPC algorithm, which is implemented in the Phase-II of proposed SSEPC. The Algorithm 1 invokes Algorithm 2, Algorithm 3, Algorithm 4, and Algorithm 5 as different modules as and when required.

Algorithm 2 deliberates the pseudo code for searching the appropriate cloud region cr_x. Each region is a distinguished geographic area of the cloud datacenters on earth, which is independent of the other regions. The pseudo code for searching the appropriate availability zone az_y is given in Algorithm 3. Here, our proposed algorithm does the exploration to find out the suitable region and the availability zone with utmost care for the maximization of green energy utilization and minimization of overall response time.

Algorithm 4 represents the pseudo code for searching the appropriate datacenter d_i. These datacenters are the physical infrastructures of the cloud providers. The process of exploitation in the first phase of SSEPC is executed to find out the appropriate datacenter for VMP. The VM request from the user reached at the availability zone is mapped to the datacenter having maximum amount of green energy, provided the average response time should not exceed threshold value. If sufficient green energy is not available in any of the datacenters, then the brown energy is taken into consideration. A list is maintained and updated on regular intervals to monitor the consumption of energy by the datacenters. The list contains the datacenters in increasing order of energy consumption. The datacenter consuming minimum energy is chosen for VMP, provided the average response time should not exceed the threshold value. Else the datacenter has been chosen based on the distance of the user from the datacenter to minimize the response time.

Algorithm 5 portrays the second phase of proposed SSEPC on the basis of emperor penguins colony algorithm to search the most suitable server s_j in a datacenter to host the virtual machine. Servers accommodate the VMs till they have not reached the upper bound threshold value of a server's capacity. Servers having less number of VMs, where server capacity falls below the lower bound, let the VMs migrate to other servers and go to sleep mode for conserving energy. Even the servers crossing the upper bound of their capacity while accommodating the VMs, let the excess VMs migrate to other servers to maintain the balance of load on each server with respect to their capacity. In this way we have implemented the VM migration technique for considering the running VMP along with the initial VMP implemented in the first phase of SSEPC. In both the phases of SSEPC, we have considered the priority of user requests in VMP for a better QoE.

The time to find the suitable cloud region among the set of u regions is $\Theta(ulogu)$. Similarly, the time for finding an appropriate availability zone in a cloud region out of the v numbers of availability zones is $\Theta(vlogv)$. The time complexity of searching a datacenter in an availability zone and then searching a server in a particular datacenter are $\Theta(mlogm)$ and $\Theta(nlogn)$ respectively due to the m number of datacenters in an availability zone and n number of servers in a datacenter. Hence, the time complexity to find a region globally, an availability zone in the region, a datacenter therein, and a server in the datacenter to place a VM request from a user is $\Theta(ulogu+vlogv+mlogm+nlogn)$.

5. Performance Evaluation

This section illustrates the setup for the experimental work for the evaluation of the proposed algorithm and its counterparts. The results obtained from the experiments are analyzed and presented as graphs to validate the efficiency of the proposed methodology in comparison to some other recent metaheuristic techniques for VMP.

5.1. Experimental Setup

An Infrastructure-as-a-Service (IaaS) cloud environment is required to evaluate the proposed methodology. As it is difficult to evaluate on real cloud infrastructure, the experiments have been conducted using CloudSim [35]. To simulate the work, three AWS cloud regions, i.e. US East (N. Virginia), US West (N. California), and Europe (Ireland) are considered with codes us-east-1, us-west-1, and eu-west-1 respectively [9]. The availability zone is coded as its region code followed by a letter for identification, e.g. us-east-1a [9]. Availability zones are mapped independently to each AWS account, i.e. us-east-1a availability zone for one AWS account may not be the same physical location in a region as us-east-1a for another AWS account. Here, the datacenters are considered of comprising four numbers of servers or the hosts with dissimilar configurations. Each host encompasses some cores of processing, speed of processing, bandwidth, RAM, and storage. 300 VMs with disparate specifications are deliberated for mapping onto the servers. The VM specifications are portrayed in the Table 3. This research considers a heterogeneous cloud milieu, where the count of servers concurrently vary with the count of VMs to endorse the efficacy of the proposed metaheuristic technique.

The QoS parameters such as the energy consumption, carbon emission, response time, and SLA violations are taken into consideration for validating the proposed metaheuristic

Algorithm 1 Salp Swarm based Emperor Penguins Colony (SSEPC) algorithm for the virtual machine placement

Require: The set of cloud regions CR with size u, the list AZ of v number of availability zones in each region, set of datacenters D of size m in each availability zone, list of servers S in each datacenter with size n, set of p number of virtual machines VM, set of q number of requests R, time interval t, response time threshold Th_R.

Ensure: Requests allocated to destination.

for r_l in R **do**
Call CloudRegion_Search()
Call AvailabilityZone_Search()
Call Datacenter_Search()
if F_{val_1} satisfies according to Eqn. (15) **then**
Allocate r_l to $cr_x \rightarrow az_y \rightarrow d_i$
The leader request in the request queue of user_account_id (r_l) = r_l
user_account_id = user_account_id (r_l)
Add r_l to user_account_id list
for r_{l+1} in R **do**
if user_account_id (r_{l+1}) == user_account_id **then**
r_{l+1} is the follower request, which will follow the search path of r_l
Allocate r_{l+1} to $cr_x \rightarrow az_y \rightarrow d_i$
Add r_{l+1} to user account id list
end if
$l = l + 1$
end for
end if
for r_l in user_account_id list **do**
Sort r_l in descending order of priority
end for
for r_l in user_account_id list **do**
for s_j in d_i **do**
if $Capi, s_j > Th_{UB}$ **then**
$j = j + 1$
else if $Capi, s_j < Th_{UB}$ **then**
$j = j + 1$
else
if F_{val_2} satisfies according to Eqn. (16) **then**
Allocate r_l to s_j
else
$j = j + 1$
end if
end if
end for
end for
Update R by removing all r_l allocated to servers
Call VM_Migration()
end for

Algorithm 2 Algorithm of **CloudRegion_Search( )**

Require: The set CR of cloud regions cr_x with size u, the green energy value of a cloud region $E^G_{cr_x}$, the set R of requests r_l, time interval t, threshold of response time Th_R. If $cr_x$is 100% enabled with green energy, then let us assume $E^G_{cr_x} = 1$. Let the response time for request r_l allocated to cr_x is T_{l,cr_x}.
Ensure: Request r_l is allocated to appropriate cloud region cr_x.

```
Generate the list of regions CR accessible to r_l
for cr_x in CR do
    if E^G_{cr_x} ≥ 0.5 then
        Add cr_x to the sorted list CR^G arranged in descending order based on E^G_{cr_x}
    else
        Add cr_x to the sorted list CR^D organized in rising order based on distance between the
        region and the user
    end if
end for
if CR^G ≠ φ then
    Allocate r_l to the cr_x in CR^G such that T_{l,cr_x} ≤ Th_R
else
    Allocate r_l to the cr_x in CR^D
end if
return cr_x
```

Algorithm 3 Algorithm of **AvailabilityZone_Search( )**

Require: The set AZ of availability zones az_y with size v, the green energy value of an availability zone $E^G_{az_y}$, the set R of request r_l, time interval t, threshold of response time Th_R. If az_y is 100% enabled with green energy, then let us assume $E^G_{az_y} = 1$. Let the response time for request r_l allocated to cr_x and az_y are T_{l,cr_x} and T_{l,az_y} respectively.
Ensure: Request r_l allocated to appropriate availability zone az_y.

```
Generate the list of unconstrained availability zones AZ available to r_l
for az_y in cr_x do
    if E^G_{az_y} ≥ 0.5 then
        Add az_y to the sorted list AZ^G arranged in descending order based on E^G_{az_y}
    else
        Add az_y to the sorted list AZ^D arranged in the ascending order based on the distance
        of availability zone az_y from the user
    end if
end for
if AZ^G ≠ φ then
    Allocate r_l to the az_y in AZ^G such that (T_{l,cr_x} + T_{l,az_y}) ≤ Th_R
else
    Allocate r_l to the az_y in AZ^D
end if
return az_y
```

technique SSEPC. The average response time data is taken from [36], which has been collected in two days with interval of 15 minutes through a real cluster. Facebook data [37] has been fetched and analysed to estimate daily requests from the active users, assuming

Algorithm 4 Algorithm of **Datacenter_Search( )**

Require: The set D of datacenters d_i with size m, the green energy value of a datacenter $E^G_{d_i}$, the value of consumption of energy of a datacenter E_{d_i}, the set of request r_l, time interval t, threshold of response time Th_R. If d_i is 100% enabled with green energy, then let us assume $E^G_{d_i} = 1$. Let the response time for request r_l allocated to cr_x, az_y, and d_i are T_{l,cr_x}, T_{l,az_y}, and T_{l,d_i} respectively.
Ensure: Request r_l allocated to appropriate datacenter d_i.

```
for d_i in az_y do
    if E^G_{d_i} ≥ 0.5 then
        Add d_i to the sorted list D^G organized in descending order based on E^G_{d_i}
    else if E_{d_i} < 0.5 then
        Add d_i to the sorted list D^E arranged in the increasing order of E_{d_i}
    else
        Add d_i to the sorted list D^D organized in the ascending order based on the distance
        between datacenter d_i and the user
    end if
end for
if D^G ≠ ϕ then
    Allocate r_l to the d_i in D^G such that (T_{l,cr_x} + T_{l,az_y} + T_{l,d_i}) ≤ Th_R
else if D^E ≠ ϕ then
    Allocate r_l to the d_i in D^E such that (T_{l,cr_x} + T_{l,az_y} + T_{l,d_i}) ≤ Th_R
else
    Allocate r_l to the d_i in D^D
end if
return d_i
```

Table 3. Virtual Machine Specification

Type of CPU	Number of Cores for Processing	Speed in MIPS	RAM in GB	Storage in GB	Bandwidth in MIPS
Core_i7_Extreme_ Edition	1-4	500	2	20	1024

that one request is submitted per user on daily basis. The count of requests are estimated in millions from various time zones. The carbon intensity data of the considered cloud locations has been collected on hourly basis obtained on 9th January 2019 [11]. In this work, the solar power is considered as the green energy, whose data from different time zones for the considered cloud locations has been collected [38].

For evaluating the efficacy of SSEPC VMP algorithm, we have chosen OEMACS, GATA, and SCA-SSA as three recent hybrid algorithms of VM placement proposed by Liu et al. [12], Zhao et al. [13], and Gharehpasha et al. [15] respectively for comparison with the proposed methodology. As these three algorithms are evolved by the hybridization of two other existing algorithms just like the hybridization applied in this proposed work, we have chosen them as the appropriate contemporaries in the same direction for comparison.

Algorithm 5 Algorithm of **VM_Migration( )**

Require: The set S of servers s_j with size n, the set VM of virtual machines vm_k with size p, time interval t, upper threshold of server capacity Th_{UB}, and lower threshold of server capacity Th_{LB}.
Ensure: Runtime VM migration for VMP.

for s_j in d_i **do**
  if $Cap_{i,s_j} < Th_{LB}$ **then**
    for vm_k in s_j **do**
      Insert vm_k to vm_migration list in descending order of priority
    end for
    s_j goes to sleep() mode
  else if $Cap_{i,s_j} > Th_{UB}$ **then**
    for vm_k in s_j **do**
      if vm_k is requesting for scaling up resources **then**
        Insert vm_k to vm_migration list in descending order of priority
      end if
    end for
  else
    $j = j + 1$
  end if
end for
for vm_k in vm_migration list **do**
  for s_j in d_i **do**
    if $Th_{LB} \leq Cap_{i,s_j} \leq Th_{UB}$ **then**
      if F_{val_2} satisfies according to Eqn. (16) **then**
        Allocate vm_k to s_j
      else
        $j = j + 1$
      end if
    else
      $j = j + 1$
    end if
  end for
end for

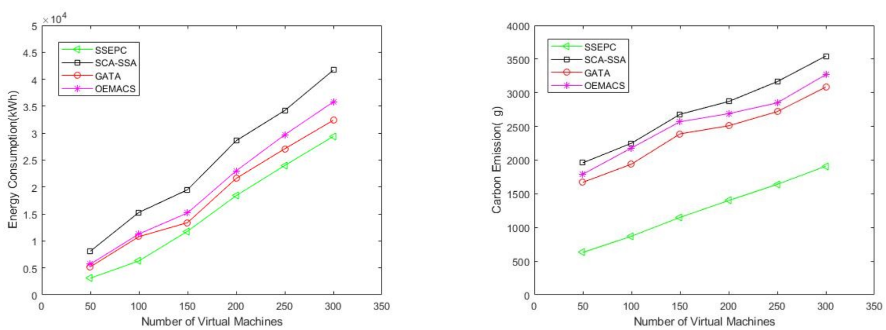
5.2. Obtained Results and Discussion

After simulating the experiments in CloudSim, the results are obtained for validating this work. The results obtained are analysed and are portrayed as graphs in this sub-section.

In SCA-SSA and OEMACS, out of the two participating algorithms of hybridization, one algorithm, i.e. SCA and ACS respectively are conducting the entire searching mechanism, although the search space broadens in this experiment. In GATA, the genetic algorithm takes the charge of searching, where tabu search helps in improving the local search ability of GA as an integrated method. These three algorithms are found to be slowing down in performance with the increase of the search space. In contrast, the proposed methodology SSEPC divides the entire search space into two parts and implements two algorithms in multi-stage collaborative approach for them separately by excelling in performance.

Figure 2(a) shows, SSEPC consumes less energy as compared to SCA-SSA, GATA, and OEMACS. The values of average energy consumption while implementing SSEPC, GATA, OEMACS, and SCA-SSA are found to be $1.48 \times (10)^4$ KWh, $1.7 \times (10)^4$ KWh, $1.89 \times (10)^4$ KWh, and $2.31 \times (10)^4$ KWh respectively. It is already stated above that SSEPC is more efficient in searching than others, for which it avails most suitable resources with better utilization and implements the runtime VMP too. So, it produces optimum result in energy consumption as compared to other algorithms.

Figure 2(b) portrays, SSEPC emits less carbon as compared to its counterparts, i.e. 1230g, 2382g, 2543g, and 2728g of carbon footprints are emitted on an average in case of SSEPC, GATA, OEMACS, and SCA-SSA respectively. SSEPC gives priority to the green energy availability while mapping VMs to the appropriate resources. Each datacenter runs on green energy till its availability and then switches to brown energy. As SCA-SSA, GATA, and OEMACS have not considered the renewable energy sources to operate the datacenters, they are emitting more carbon.



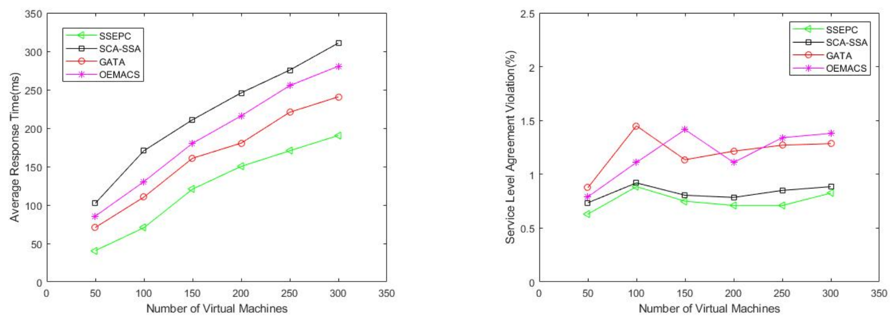
(a) Comparison of Energy Consumption of SSEPC with other VMP Algorithms

(b) Comparison of Carbon Emission of SSEPC with other VMP Algorithms

Fig. 2. Comparison of Energy Consumption and Carbon Emission of SSEPC with others

Due to prioritization of user requests during runtime VMP in SSEPC, along with the implementation of leader-follower mechanism, the proposed algorithm gives optimum average response time with a better QoE for the users than the other metaheuristic algorithms considered for comparison. This can be observed in Figure 3(a). The average response time during the implementation of SSEPC, GATA, OEMACS, and SCA-SSA are obtained as 117.32ms, 157.48ms, 182.15ms, and 208ms respectively.

GATA and OEMACS have not taken care of SLA violations as a QoS parameter. Although SCA-SSA has considered the minimization of SLA violations during VMP, it slows down in a larger search space resulting a bit higher SLA violations as compared to SSEPC. Figure 3(b) depicts the comparison of SLA violations of SSEPC and others. From Figure 3(b), it can be viewed that SSEPC minimizes the SLA violations more in comparison to others. The average SLA violations in percentage for SSEPC, SCA-SSA, GATA, and OEMACS are found to be 0.8%, 0.81%, 0.83%, and 0.87% respectively.



(a) Comparing Average Response Time of SSEPC with other VMP Algorithms

(b) Comparison of SLA Violation of SSEPC with other VMP Algorithms

Fig. 3. Comparison of Average Response Time and SLA Violation of SSEPC with others

The results obtained clearly indicate that the proposed SSEPC outperforms its counterparts for the QoS parameters like energy consumption, carbon emission, average response time, and SLA violations. To keep the brevity of this paper, we have reduced the detailed description in this section.

6. Conclusion and Future Works

This work proposes a multi-stage collaborative hybrid algorithm for the solution of VMP problem in the cloud milieu. In this work, the search space is broaden by the incorporation of cloud regions and the availability zones along with the datacenters and the servers to make it more realistic. The proposed SSEPC methodology explores in the first phase to search the appropriate region as well as the availability zone and then exploits to search the suitable datacenter therein for the placement of the VMs by utilizing salp swarm optimization. In the second phase, it implements emperor penguins colony algorithm to search locally in a datacenter to find the most suitable server to place the VMs. Along with initial VMP, runtime VMP is also applied in this work. Some vital QoS metrics like energy consumption, carbon emission, average response time, and SLA violations are considered to appraise the efficacy of the recommended hybrid metaheuristic technique in comparison with some recent VMP algorithms, where it outperforms others. Utilization of green energy is prioritized during the process of VMP for the environmental sustainability. User requests are processed on the basis of priority for better QoE of the users.

As part of our future work, we plan to extend our approach to include the cost of energy consumption and carbon footprint cost along with calculating their amounts to analyze the evaluation from both qualitative and quantitative perspective. Hybridization of some other techniques in this direction can be explored to optimize the QoS metrics better. The exploration of adaptive algorithms can be considered to learn and predict the optimal VMP over time, considering energy consumption, carbon footprint, and other cloud service demands.

References

1. Parida, S., Pati, B., Nayak, S. C., Panigrahi, C. R., Weng, T. H.: PE-DCA: Penalty elimination based data center allocation technique using guided local search for IaaS cloud. Computer Science and Information Systems, 19(2), 679-707. (2022)
2. Feng, H., Deng, Y., Li, J.: A global-energy-aware virtual machine placement strategy for cloud data centers. Journal of Systems Architecture, 116, 102048. (2021)
3. Wikipedia, Data center (2023).[Online]. Available: https://en.wikipedia.org/wiki/Data-center (current June 2023)
4. Koot, M., Wijnhoven, F.: Usage impact on data center electricity needs: A system dynamic forecasting model. Applied Energy, 291, 116798, 12-27. (2021)
5. Sarpong, K. A., Xu, W., Gyamfi, B. A., Ofori, E. K.: A step towards carbon neutrality in E7: The role of environmental taxes, structural change, and green energy. Journal of Environmental Management, 337, 117556. (2023)
6. Xu, M., Buyya, R.: Managing renewable energy and carbon footprint in multi-cloud computing environments. Journal of Parallel and Distributed Computing, 135, 191-202. (2020)
7. Abbasi-khazaei, T., Rezvani, M. H.: Energy-aware and carbon-efficient VM placement optimization in cloud datacenters using evolutionary computing methods. Soft Computing, 26(18), 9287-9322. (2022)
8. AWS, Regions and Availability Zones. [Online]. Available: https://aws.amazon.com/about-aws/global-infrastructure/regions_az/ (current June 2023)
9. AWS, Amazon Elastic Compute Cloud, Regions and Zones (2023). [Online]. Available: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html (current June 2023)
10. Mirjalili, S., Gandomi, A. H., Mirjalili, S. Z., Saremi, S., Faris, H., Mirjalili, S. M.: Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. Advances in engineering software, 114, 163-191. (2017)
11. Harifi, S., Khalilian, M., Mohammadzadeh, J., Ebrahimnejad, S.: Emperor Penguins Colony: a new metaheuristic algorithm for optimization. Evolutionary Intelligence, 12(2), 211-226. (2019)
12. Liu, X. F., Zhan, Z. H., Deng, J. D., Li, Y., Gu, T., Zhang, J.: An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE transactions on evolutionary computation, 22(1), 113-128. (2016)
13. Zhao, D. M., Zhou, J. T., Li, K.: An energy-aware algorithm for virtual machine placement in cloud computing. IEEE Access, 7, 55659-55668. (2019)
14. Samriya, J. K., Chandra Patel, S., Khurana, M., Tiwari, P. K., Cheikhrouhou, O.: Intelligent SLA-aware VM allocation and energy minimization approach with EPO algorithm for cloud computing environment. Mathematical Problems in Engineering. (2021)
15. Gharehpasha, S., Masdari, M., Jafarian, A.: Power efficient virtual machine placement in cloud data centers with a discrete and chaotic hybrid optimization algorithm. Cluster Computing, 24(2), 1293-1315. (2021)
16. Zhao, D., Zhou, J.: An energy and carbon-aware algorithm for renewable energy usage maximization in distributed cloud data centers. Journal of Parallel and Distributed Computing, 165, 156-166. (2022)
17. Xu, M., Toosi, A. N., Buyya, R.: Ibrownout: an integrated approach for managing energy and brownout in container-based clouds. IEEE Transactions on Sustainable Computing, 4(1), 53-66. (2018)
18. Tang, M., Pan, S.: A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural processing letters, 41(2), 211-221. (2015)
19. Yadav, R., Zhang, W., Kaiwartya, O., Singh, P. R., Elgendy, I. A., Tian, Y. C.: Adaptive energy-aware algorithms for minimizing energy consumption and SLA violation in cloud computing. IEEE Access, 6, 55923-55936. (2018)

20. Khodayarseresht, E., Shameli-Sendi, A., Fournier, Q., Dagenais, M.: Energy and carbon-aware initial VM placement in geographically distributed cloud data centers. Sustainable Computing: Informatics and Systems, 100888. (2023)
21. Belabed, D., Secci, S., Pujolle, G., Medhi, D.: Striking a balance between traffic engineering and energy efficiency in virtual machine placement. IEEE Transactions on Network and Service Management, 12(2), 202-216. (2015)
22. Justafort, V. D., Beaubrun, R., Pierre, S.: A hybrid approach for optimizing carbon footprint in intercloud environment. IEEE Transactions on Services Computing, 12(2), 186-198. (2016)
23. Anusooya, G., Vijayakumar, V.: Reduced carbon emission and optimized power consumption technique using container over virtual machine. Wireless Networks, 27, 5533-5551. (2021)
24. Ting, T. O., Yang, X. S., Cheng, S., Huang, K.: Hybrid metaheuristic algorithms: past, present, and future. Recent advances in swarm intelligence and evolutionary computation, 71-83. (2015)
25. Microsoft, Azure, Regions and availability Zones (2023). [Online]. Available: https://docs.microsoft.com/en-us/azure/availability-zones/az-overview (current June 2023)
26. Forbes, With The Public Clouds Of Amazon, Microsoft And Google, Big Data Is The Proverbial Big Deal (2017). [Online]. Available: https://www.forbes.com/sites/johnsonpierr/2017/06/15/with-the-public-clouds-of-amazon-microsoft-and-google-big-data-is-the-proverbial-big-deal/?sh=4ba4ae7c2ac3 (current June 2023)
27. Arcaini, P., Riccobene, E., Scandurra, P.: Modeling and analyzing MAPE-K feedback loops for self-adaptation. In 2015 IEEE/ACM 10th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (pp. 13-23). IEEE. (2015)
28. Basmadjian, R., Ali, N., Niedermeier, F., De Meer, H., Giuliani, G.: A methodology to predict the power consumption of servers in data centres. In Proceedings of the 2nd international conference on energy-efficient computing and networking (pp. 1-10). (2011)
29. Wang, N., Zhang, J., Xia, X.: Energy consumption of air conditioners at different temperature set points. Energy and Buildings, 65, 412-418. (2013)
30. Tang, Q., Gupta, S. K. S., Varsamopoulos, G.: Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach. IEEE Transactions on Parallel and Distributed Systems, 19(11), 1458-1472. (2008)
31. National grid, What is carbon intensity? (2023). [Online]. Available: https://www.nationalgrid.com/stories/energy-explained/what-is-carbon-intensity (current June 2023)
32. Babu, D. LD., Venkata Krishna, P.: Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl. Soft Computer Journal, 13(5), 2292-2303. (2013)
33. Rizk-Allah, R. M., Hassanien, A. E., Elhoseny, M., Gunasekaran, M.: A new binary salp swarm algorithm: development and application for optimization tasks. Neural Computing and Applications, 31(5), 1641-1663. (2019)
34. Son, J., Buyya, R.: Priority-aware VM allocation and network bandwidth provisioning in software-defined networking (SDN)-enabled clouds. IEEE Transactions on Sustainable Computing, 4(1), 17-28. (2018)
35. Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A., Buyya, R.: CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23-50. (2011)
36. Doyle, J., Shorten, R., O'Mahony, D.: Stratus: Load balancing the cloud for carbon emissions control. IEEE Transactions on Cloud Computing, 1(1), 1-1. (2013)
37. Atikoglu, B., Xu, Y., Frachtenberg, E., Jiang, S., Paleczny, M.: Workload analysis of a large-scale key-value store. In Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE joint international conference on Measurement and Modeling of Computer Systems (pp. 53-64). (2012)
38. E. Commission, Photovoltaic Geographical Information System (2022). [Online]. Available: https://re.jrc.ec.europa.eu/pvg_tools/en/tools.html (current June 2023)

Bivasa Ranjan Parida received the B.E. degree in CSE in 2004, and the M.Tech. degree in CSE in 2010 from Biju Patnaik University of Technology, Odisha, India. Currently, he is pursuing his Ph.D. in CSE as a scholar in Veer Surendra Sai University of Technology, Burla, India under the guidance of Prof. Amiya Kumar Rath. His research area includes cloud computing, grid computing, and wireless body area network. He has 18 years of teaching and research experience.

Amiya Kumar Rath received the B.E. degree in CSE from Dr. Babasaheb Ambedkar Marathwada University, Aurangabad in 1990, the M.B.A. degree in Systems Management from Shivaji University in 1993, the M.Tech. degree in Computer Science from Utkal University in 2001, and the Ph.D. degree in Computer Science from Utkal University in 2005, with a focus on embedded systems. He is currently a Professor with the Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, India. He has contributed over 80 research level papers to many national and international journals and conferences. He has published seven books by reputed publishers. His research interests include embedded systems, ad hoc networks, sensor network, power minimization, evolutionary computation, and data mining. Currently, he has been deputed as Vice-Chancellor of Biju Patnaik University of Technology, Odisha, India.

Bibudhendu Pati is an Associate Professor in the Department of Computer Science at Rama Devi Women's University, Bhubaneswar, India. He received his BE in CSE degree with Honours, ME in CSE from NITTTR, Chandigarh, PhD degree from IIT Kharagpur, India. He has around 25 years of experience in teaching and research. His current research interests include Wireless Sensor Networks, Mobile Cloud Computing, Internet of Things, and Advanced Network Technologies. He has been involved in many professional and editorial activities. He has got several papers published in reputed journals, conference proceedings, and books of International repute. He is the Life Member of ISTE, Life Member of CSI, and Senior Member of IEEE.

Chhabi Rani Panigrahi is an Assistant Professor in the Department of Computer Science at Rama Devi Women's University, Bhubaneswar, India. Prior to this she was associated with Central University of Rajasthan. She received her PhD degree in Computer Science & Engineering from Indian Institute of Technology Kharagpur, India. Her current interests include software testing, mobile cloud computing, Internet of Things, and machine learning. She has more than 23 years of teaching and research experience. She has published several scholarly articles in journals and conferences of international repute. She was invited to organize and chair many international conferences. She is the author of a text book and a few edited books. She is a life member of the Indian Society of Technical Education (ISTE) and a member of IEEE and the Computer Society of India (CSI).

Hitesh Mohapatra received the B.E. degree in Information Technology in 2006, and the M.Tech. degree in CSE in 2009 from Biju Patnaik University of Technology, Odisha, India. He received his Ph.D. in Computer Science & Engineering in 2021 from Veer Surendra Sai University of Technology, Burla, India. He has contributed 20 SCI and Scopus indexed research papers, 15 international/ national conferences and 2 books on Software Engineering and C Programming respectively. He has 12 years of teaching experience both in industry and academia. His research interests include wireless sensor network, smart city, smart grid and smart water.

Tien-Hsiung Weng is currently working as a Professor in the Department of Computer Science and Information Engineering, Providence University, Taichung City, Taiwan. He received his Ph.D. in Computer Science from the University of Houston, Texas. His research interests include parallel computing, high performance computing, scientific computing, and machine learning.

Rajkumar Buyya is a Fellow of IEEE, Professor of Computer Science and Software Engineering and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft, a spin-off company of the University. He is one of the highly cited authors in computer science and software engineering worldwide (h-index=109, g-index=230, 58,300+ citations). Microsoft Academic Search Index ranked him as #1 author in the world (2005-2016) for both field rating and citations evaluations in the area of Distributed and Parallel Computing." A Scientometric Analysis of Cloud Computing Literature" by German scientists ranked him as the World's Top-Cited (#1) Author and the World's Most-Productive (#1) Author in Cloud Computing. Software technologies for Grid and Cloud computing developed under his leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Manjrasoft's Aneka Cloud technology developed under his leadership has received "2010 Frost & Sullivan New Product Innovation Award". Recently, he received "Mahatma Gandhi Award" along with Gold Medal for his outstanding and extraordinary achievements in Information Technology field and services rendered to promote greater friendship and India-International cooperation. He served as the founding Editor-in-Chief of the IEEE Transactions on Cloud Computing. He is currently serving as Co-Editor-in-Chief of Journal of Software: Practice and Experience, which was established over 45 years ago.

Received: September 23, 2023; Accepted: January 3, 2024.