Research Article

Chaotic Salp Swarm Optimization-Based Energy-Aware VMP Technique for Cloud Data Centers

S. Parthiban, 1 A. Harshavardhan, 2 S. Neelakandan, 3 Vempaty Prashanthi, 2 Abdul-Rasheed Akeji Alhassan Alolo, 4 and S. Velmurugan 5

1Department of Computer Science and Engineering, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, India
2Department of Computer Science and Engineering, VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India
3Department of Computer Science and Engineering, R.M.K. Engineering College, Chennai, India
4Department of Marketing and Corporate Strategy, Tamale Technical University, Tamale, Ghana
5Department of Computer Science and Engineering, Vel Tech Multi Tech Dr. Rangarajan Dr. Sakunthala Engineering College, Chennai, India

Correspondence should be addressed to Abdul-Rasheed Akeji Alhassan Alolo; aaakeji@tatu.edu.gh

Received 29 March 2022; Revised 23 April 2022; Accepted 28 April 2022; Published 12 May 2022

Copyright © 2022 S. Parthiban et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The amount of energy required by Cloud Data Centers (CDCs) has increased significantly in this digital age, and as a result, there is a pressing need to reduce CDC energy ingesting. Consolidation of virtual machines (VMs) and effective virtual machine placement (VMP) techniques are commonly employed in large data middles to reduce energy consumption. The VMP is an NP-hard subject with infeasible optimum explanations even for tiny data middles, and it is dealt with using the Metaheuristic Optimization Algorithm, which is an experiential approach to optimization. With this in mind, this study introduces a novel energy-aware VMP technique for CDCs that is founded on the Disordered Salp Swarm Optimization Algorithm (EAVMP-CSSA) and is enhanced for energy efficiency (EAVMP-CSSA). The EAVMP-CSSA technique attempts to reduce CDC energy ingesting by dropping the quantity of active servers supporting virtual machines. The recommended EAVMP-CSSA strategy also aims to balance the resource operation of active servers (i.e., CPU, RAM, and Bandwidth), hence reducing waste and increasing efficiency. Furthermore, by combining the ideas of chaotic maps with the standard Salp Swarm Optimization Algorithm (SSA), the CSSA is intended to improve overall performance and reduce computational costs (SSA). A comprehensive range of experimental analyses are performed to ensure that the EAVMP-CSSA technique performs better, and the findings are compared to current VMP techniques. The EAVMP-CSSA approach achieves an effective outcome with a maximum service rate of 98.12%, whereas the Random, FFD, ACO, and AP-ACO procedures achieve a minimum service rate of 74.40%, 78.80%, 90.70%, and 96.31%, respectively. The experimental results demonstrate that the EAVMP-CSSA approach outperforms other assessment metrics.

1. Introduction

Cloud Computing (CC) is one of the effective computing modules that delivers and hosts a broad variety of services via Internet. Several enterprises are based on cloud framework instead of in-house framework for the benefits provided by cloud platforms like removing maintenance burden, attaining on demand scalability, and pay as you go pricing module [1]. The data center is utilized in the cloud platforms for providing cloud services that consumes a huge amount of energy for its processes. An overview of CC perfect is shown in Figure 1. The information center is generally equipped with huge number of physical attendants. Nearly 60% of the overall liveliness ingesting in information midpoint originate from the IT framework that is managed using Physical Machines (PMs). Thus, reducing the number of lively PMs in a data center would greatly enhance the less vigor operation rate. Virtualization is one of the
advanced techniques where the CC assets are given to clients through limitless amount of Virtual Machine (VM) depending upon a group of Service Level Agreements (SLAs) among cloud customers and providers [2]. Virtualization performs a critical part in attaining energy efficiency and high server consumption as numerous VMs are assigned to the same corporeal attendant. In a virtualized cloud platform, a sufficient quantity of active attendants is placed based on the VM deployment [3]. Henceforth, utilizing an effective Virtual Machine Placement (VMP) method could attain major impact on a data centers’ power utilization.

The VMP problem is similar to the resource allocation problems, concentrating on how to assign physical resources of PM to VM of users with their needs. The main problem in network virtualization is that the VMP problem has attained substantial interest recently. The multidimensional physical resource of the server in data center includes storage resources (Storage), computing resources (CPU), and memory resources (Memory) [4]. To assure the quality of service, it is essential that the server must have adequate multidimensional assets for the application it transports [5]. Through an additional development in transmission methods, it has increasingly different applications [6]. Particularly, memory-intensive, I/O-intensive, and computation-intensive applications coexist on the Internet recently. Apparently, the multi-resource requirements vary from application to application. The computationally intensive task is starving for huge CPU prerequisite; however, it demands only small memory.

The VMP method attempts to discover optimum allocations of VM over PM to attain their objectives of the design [7]. Several design objectives are considered in the survey, for example, reducing SLA violation, improving the power utilization, optimizing resource consumption, and so on. The VMP problematic can be considered as an NP-hard optimization problematic. Thus, various metaheuristic methods like Biogeography-Based Optimization (BBO), Glowworm Swarm Optimization (GSO), and Firefly Algorithm (FA) are applied for generating effective solutions within the moderate time [8]. Though, numerous VMP techniques [9, 10] are presented for addressing the energy utilization optimization problem in cloud platforms; they ensure well-adjusted utilization of multidimensional assets amongst dynamic PMs. These methods might allocate distinct number of residual assets for all resource types of PM. In expectation with upcoming demands, the resource left on every PMs must be balanced. Otherwise, the unbalanced residual resources might be avoided by the other VMPs, which leads to wastage of computing resource.

This study designs VMP approach using Chaotic Salp Swarm Optimization Algorithm (EAVMP-CSSA) for Cloud Data Centers (CDCs). The design of CSSA for the optimal placement of VMs shows the novelty of the work. The EAVMP-CSSA technique intends to effectively utilize the energy at the CDCs by reducing the active server count hosting VMs and turning off the idle ones. The planned EAVMP-CSSA technique derives a Fitness Function to achieve minimization of the complete liveliness ingesting of the cloud information servers with minimal resource wastage. Furthermore, it accomplishes the balanced utilization of multiple resources of the active sources to minimize the resource wastage. To highlight the improved

Figure 1: Overview of cloud computing.
performance of the proposed EAVMP-CSSA technique, a set of simulations are performed and the results are investigated under different aspects.

2. Prior Works on VMP Techniques

In Ramalingam and Mohan’s work [11], a novel method is presented in accordance with the integration of hybrid optimization algorithms for optimum deployment in CDCs. The initial impartial of the projected method is to decrease the influence utilization of CDCs during the reduction of active PMs. The next aim is to reduce the resource consumption and manage resources by an optimum deployment of VM on PM in the CDC. Nahavi et al. [12] presented a multiobjective VMP system (consider VM as a fog task) for the ECDC is known as TRACTOR that uses an ABC optimization method for energy aware assignments of VM on PM. The projected system aims to reduce the network traffic of the related VMs and energy exploitation during the data center changes and PMs.

Alboaneen et al. [13] planned a novel metaheuristic technique for enhancing Joint Task Scheduling and VMP (JTSVMP) in CDC. The JTSVMP problems consist of two processes, namely, VM placement and task scheduling, that are processed as a combined problematic to be resolved by MOA method. The projected cooptimization procedure purposes to allocate task for the VM that has the minimum performance cost within the fixed limits and later deploy the chosen VM on maximum used PH within capability limit. Gharehpasha et al. [14] projected a novel method with an integration of the SCA and SSA as separate multiobjective and disordered function for a best VMP. The initial purpose of the projected method is to decrease the control utilization in CDCs by reducing the number of lively PMs.

In Abdel-Basset et al.'s work [15], a bandwidth-conscious VMP method has been presented based on the enhanced WOA hybridized by a novel BWAP. The projected study concentrates on improving the bandwidth when the other significant aspects like CPU and memory utilization are not considered. In addition, the power utilization optimization problem was not tackled. At last, Alreshedi et al. [16] presented a cross multiobjective VMP method based on SSA and SCA. The presented method is intended to enhance the SLA violation, MTBHS, and power utilization. The presented method is related to various metaheuristics, and the attained result supports its dominance. But the bandwidth has not been utilized while illustrating PMs and VMs. Additionally, the balanced utilization of multidimensional resources in physical server was not assured.

Wei et al. [17] presented an energy-effective VMP system that continued to decrease the control utilization and transmission cost on circulation conscious datacenter network. For solving this optimization problem, an enhanced ACO using adaptive variable set was proposed for balancing its strong searching ability and fast convergence. Torre et al. [18] presented a multi-impartial technique for active VMP that examines the live relocation method for concurrently improving the overcommitment ratio, migration energy, and resource waste. This optimization method utilizes a new evolution meta-experiential approach founded on the key populace method for approximating the Pareto-optimum set of VMP using better diversity and accuracy.

In Abohamama and Hamouda’s work [19], a hybrid VMP method is presented in accordance with permutation-based GA and multidimensional supply conscious best appropriate distribution approach. The projected VMP method purposes at the minimum liveliness utilization amount of cloud datacenters by reducing the number of active servers that hosts VMs. In addition, the suggested VMP method tries to attain balanced application of multidimensional resources (Bandwidth, RAM, and CPU) of lively server that consecutively reduces the resource wastage. Wei et al. [20] balance the multiple resource application for alleviating resource fragmentation when increasing the service rate for VMP, thus avoiding excess of physical resource. For solving this biobjective optimization problem, they presented a joint bin packing heuristic and GA that attains an accurate optimum solution at low time complexity. Reducing the power cost and maintaining the QoS assurance are the two major objectives of this research [2, 18]. To effectively tackle this issue, the presented VM merging method reflects the present and upcoming consumption of possessions by the host Underload Detection (UP-PUD) and host Overload Detection (UP-POD) [19, 21]. The upcoming resource consumptions are precisely forecasted using Gray Markov-based method.

3. Background Information and Problem Statement

This section discusses the background details of PMs and VMs. Besides, the problem statement of the proposed model is as follows.

3.1. Physical Machines (PMs). The data center contains m PMs \( P = \{ p_1, \ldots, p_m \} \). A resource capacity vector \( \text{CV} \) (\( f_j \)) defines every PM \( p \in P \), whereas each dimension \( k \in [1, v] \) denotes the capability of all PM physical resources \( r_k \) in the set \( R = \{ r_1, \ldots, r_v \} \). In a usual Cloud situation, \( R = \{ \text{CPU}, \text{memory}, \text{disk}, \text{network} \} \), abstracted using the virtualization technique [18]. This research emphasizes on memory and CPU, the most committed resource in data center that affects the VM migration [22, 23].

3.2. Virtual Machines (VMs). They recognize dual groups of VMs that contribute to the deployment procedure. The received VMs are the novel VM, which increases the application or generate novel application placements. The hosted VM is now the running one [24, 25]. Together, they determine a set VM = \( \{ vm_{1}, \ldots, vm_{n} \} \) deployed on enhanced subsets of PM \( P_{\text{used}} \subseteq P \). All \( vm \in VM \) contains two \( v \)-dimensional vectors. Resource size vector \( \overline{SV} \) (\( vm \)) = \( (s_1, \ldots, s_v) \) denotes the quantity \( S_k \) of resource \( r_k \) requested by the VM \( vm \) using \( k \in [1, v] \). Resource demand vector \( \overline{DV} \) (\( vm, t \)) = \( (d_1(t), \ldots, d_v(t)) \) determines the \( vm \) task
demands $d_k(t)$ for every resource $r_k$ at time instance $t$, through $k \in [1, v]$.

3.3. Problem Statement. EAVMP The VSBPP could be summarized as follows: assume an established of inseparable substances by specific masses and established of containers through parameter size (or type), pack the entire items to the quantity of bins; thus, the amount of wasted space of the utilized bin is reduced. In this work, VMs and PHs are signified by the tierce greatest important capitals such as system bandwidth, the CPU, and memory [19, 26]. Given that “$N$” VMs and “$M$” PHs and the overall demand for the VM are lesser compared to the overall capabilities of PH. All VMs should contain sufficient resources for the allocated VM are lesser compared to the overall capabilities of PH. All VMs should be exactly allocated to one PM ((2) and (4)). All PMs should contain sufficient resources for the allocated VMs ((5)–(7)). $v_{cpu}$, $v_{mem}$, and $v_{BW}$ represent the network bandwidth, the CPU, and memory demand of $VM_i$, respectively. $p_{cpu}$, $p_{mem}$, and $p_{BW}$ denote the network bandwidth, the CPU, and memory capabilities of PH, respectively. The overall procedure of VMP is publicized in Figure 2.

The VMP problems could be equated as VSBPP.

\[
\text{Minimize } \sum_{j=1}^{M} c_j y_j, \tag{1}
\]

subjected to

\[
X_{ij} = \begin{cases} 
1, & \text{if server PH}_j \text{ is allocated to } VM_i, \\
0, & \text{Otherwise,}
\end{cases} \tag{2}
\]

\[
y_j = \begin{cases} 
1, & \text{if } \sum_{i=1}^{N} x_{ij} \geq 1, \\
0, & \text{Otherwise,}
\end{cases} \tag{3}
\]

\[
\sum_{j=1}^{M} x_{ij} = 1, \tag{4}
\]

\[
\forall j, \sum_{i=1}^{N} v_{cpu} \times x_{ij} \leq p_{cpu}, \tag{5}
\]

\[
\forall j, \sum_{i=1}^{N} v_{mem} \times x_{ij} \leq p_{mem}, \tag{6}
\]

\[
\forall j, \sum_{i=1}^{N} v_{BW} \times x_{ij} \leq p_{BW}, \tag{7}
\]

Here, $y_j$ denotes the binary parameter that specifies whether $PH_j$ has VMs or not, $c_j$ represents the cost/wasted space of PM $PH_j$, $x_{ij}$ represents the binary parameter that specifies whether $VM_i$ is allocated to $PH_j$ or not, $N$ indicates the overall VMs, and $M$ represents the overall PMs, $i \in [1, 2, \ldots, N]$ and $j \in [1, 2, \ldots, M]$.

4. Design of EAVMP-CSSA Technique

SSA is a recently presented metaheuristic algorithm that inspires the behaviour of salps in ocean. It is a class of Salpidae similar to that of jelly fish. It forages as well as navigates in a swarm that represents salp chain. SSA is a new kind of PSO that modules the salp cable [21, 27]. The salp populace has follower and leader salps. The location of every salp in is in dimension search space, where $d$ denotes the quantity of parameters in a specific problematic, like additional group-built method. The present location course of $n$ salp in the exploration interplanetary is $X^j_l = [x^j_{l,1}, x^j_{l,2}, \ldots, x^j_{l,n}], j = 1, 2, \ldots, n$. The spearhead salp upgrades its location, and it is given by

\[
X^j_l = \begin{cases} 
F_i + C_1 \left( (u_{b, l} - ul) C_2 + lb_1 \right), & C_3 \geq 0, \\
F_i - C_1 \left( (u_{b, l} - ul) C_2 + lb_1 \right), & C_3 < 0,
\end{cases} \tag{8}
\]

where $X^j_l$ denotes the location of the spearhead salp in the $i^{th}$ measurement, $F_i$ represents food location in the $i^{th}$ measurement, and $ub_l$ and $lb_l$ characterize higher and lesser boundaries in the $i^{th}$ measurement correspondingly. $C_1$, $C_2$, and $C_3$ denote module coefficients. These coefficients are arbitrary values that are utilized for specific determinations [26, 27]. The initial coefficient $C_1$ represents the balance between exploitation and exploration that denotes the primary variable in the method. $C_1$ is determined by
\[ C_1 = 2e^{-(u/T_{\text{max}})^2}, \]  
where \( t \) denotes the present repetition and \( T_{\text{max}} \) indicates the extreme number of repetitions. \( C_2 \) and \( C_3 \) represent arbitrary values created uniformly that lies between zero and one. The follower salp updates their position based on Newton’s law of motion, and it is given by

\[ X_i^k = \frac{1}{2} (X_i^{k-1} + X_i^{k-2}) + \omega(t) \]  
where \( X_i^k \) denotes the location of \( k \)th supporter salp in the \( i \)th dimension and \( n \) represents the entire amount of salp subdivisions. The process involved in SSA is given in Algorithm 1. Population-based metaheuristic method shares different benefits that includes simplicity, scalability, and computation time reduction. But this method has two major drawbacks, namely, low convergence rate and recession in local optimal. A specific method to conquer this problem and improve the efficacy of meta experiential procedures is to place the disorder model. The disordered chart is applied for analyzing the behavior of dynamic systems using crucial characteristics through higher efficiency. Manage this parameter by using the chaotic maps that share the similar characteristics. Enhancing the efficiency of metaheuristic method with the disordered charts could be placed only for decreasing constant \( C_2 \). The value of \( C_2 \) could be substituted by the value of a suitable disordered chart at the present repetition, and it is given by

\[ C_2' = \omega(t), \]  
where \( \omega(t) \) denotes the rate of disordered chart at \( t \)th repetition. Equation (8) could be rephrased with the novel rate of \( C_2 \), and it is given by

\[ X_i^1 = \begin{cases} F_i + C_1((ub_i - ul_i)\omega(r) + lb_i), & C_1 \geq 0, \\ F_i - C_1((ub_i - ul_i)\omega(r) + lb_i), & C_1 < 0. \end{cases} \]  
The chaos model is a popular numerical method utilized for analyzing the behavior of dynamic systems using crucial primary conditions. The specific method to show this behavior by utilizing chaotic map is moreover separate or else incessant. Disordered charts could be placed only for deterministic organizations using prediction performance. At present, confusion model turns more interesting in many streams like robotics, computer science, microbiology, and physics. The chaotic map becomes the robust solution for enhancing the efficiency of metaheuristic method with the enhancement of their arbitrary variables [27–29]. This arbitrary parameter is extracted on the basis of unchanging or Gaussian delivery and hence they could be managed better by using the chaotic maps that share the similar characteristic through higher efficiency. Manage this parameter using a chaotic chart, and the local optimal is reduced, whereas the meeting is increased. The logistic chart is the optimal disordered chart for this optimizer based on the outcome of the optimization. Robert first presented the logistic chart on May 1, 1976, and it has been in use ever since. The following is the most commonly used formula for a logistic disordered map:

\[ \omega(t + 1) = a\omega(t)[1 - \omega(t)], \quad a = 4, \]  
where \( \omega(t) \) means the rate of disordered chart at \( t \)th repetition. The original disorder of the disordered charts is considered to be 0.7 (\( \omega(0) = 0.7 \)).

Provided “N” VMs and “M” PMs, the EAVMP-CSSA technique recommends various variations for the VMs, which is required to be allocated to the existing PMs. The main objective of the EAVMP-CSSA technique is to decrease the total liveliness operation of the used PMs and thereby minimize the total cost of the cloud provider. The fitness function is given as follows.

Minimize \( f(x) = \sum_{j=1}^{M} y_j \times ((P_j^{\text{busy}} - P_j^{\text{idle}}) \times U_j^{\text{cpu}} + P_j^{\text{idle}}) \),

where \( f(x) \) signifies the entire liveliness utilization of the PMs, \( y_j \) is a binary mutable that designates whether \( PH \) comprises VMs or not, \( P_j^{\text{busy}} \) is the higher energy operation of PM \( PH \), \( P_j^{\text{idle}} \) is the lower energy utilization of PM \( PH \) \( (P_j^{\text{idle}} = 0.6 \times P_j^{\text{busy}}) \), and \( U_j^{\text{cpu}} \) is the CPU operation ratio of PM \( PH \), and it is given by

\[ U_j^{\text{cpu}} = \sum_{i=1}^{N} x_{ij} \times \frac{\nu_{\text{cpu}, j}}{P_{\text{cpu}, j}}, \]

where \( x_{ij} \) is the binary parameter indicating whether \( VM \) is allocated to \( PH \) or not, \( \nu_{\text{cpu}, j} \) is the CPU demand of \( VM \) \( VM_j \), and \( P_{\text{cpu}, j} \) is the CPU volume of \( PM \) \( PH_j \).

Provided a VM which is to be allocated, the EAVMP-CSSA technique selects a PM which offers the capitals (CPU, memory, and system bandwidth) required by the VMs [7, 30–32]. The EAVMP-CSSA technique chooses the PM with low-resource wastage after allocating to the present VM. In order to completely exploit the multidimensional resources, the following equation is used to determine the wasted resources.

\[ W_j = \frac{|2L_j^{\text{cpu}} - L_j^{\text{mem}} - L_j^{\text{bw}}| + \varepsilon}{U_j^{\text{cpu}} + U_j^{\text{mem}} + U_j^{\text{bw}}}, \]

where \( W_j \) denotes the resource wastage of PM \( PH_j \), \( L_j^{\text{cpu}} \), \( L_j^{\text{mem}} \), and \( L_j^{\text{bw}} \) represent the normalized residual CPU, memory, and bandwidth of PM \( PH_j \), respectively. \( U_j^{\text{cpu}} \), \( U_j^{\text{mem}} \), and \( U_j^{\text{bw}} \) represent the regularized CPU, reminiscence, and bandwidth utilization of PM \( PH_j \) correspondingly [33]. \( \varepsilon \) is an actual unimportant hopeful real number and the value is fixed to 0.0001. The aim of (16) is to successfully utilize the multidimensional resources and stability the residual possessions on every PM along distinct extent. \( U_j^{\text{cpu}} \) is already represented in (15), while the rest of the raptors are represented as follows:
(1) Initialization of arbitrarily produced salp swarm population \( X_i (i = 1, 2, \ldots, n) \).
(2) Compute the fitness values (FV) of all salps.
(3) \( X^* \) = the best search agent.
(4) while termination criterion is unsatisfied
(5) Updating initial coefficient \( C_1 \) using (9)
(6) for every salp
(7) if \( n = 1 \)
(8) Updating the place of leader salp using (9)
(9) else
(10) Updating the location of the follower salp using (10)
(11) end if
(12) end for
(13) Determine the FV of all salps.
(14) Update \( X^* \) when better solution exists.
(15) end while
(16) obtain the optimal solution \( X^* \) and its FV.

**Algorithm 1: Pseudocode of SSA.**

\[
L_{cpu}^j = \frac{P_{cpu} - \left( \sum_{i=1}^{N} x_{ij} \times \nu_{cpu} \right)}{P_{cpu}}
\]

\[
L_{mem}^j = \frac{P_{mem} - \left( \sum_{i=1}^{N} x_{ij} \times \nu_{mem} \right)}{P_{mem}}
\]

\[
L_{BW}^j = \frac{P_{BW} - \left( \sum_{i=1}^{N} x_{ij} \times \nu_{BW} \right)}{P_{BW}}
\]

\[
U_{mem}^j = \frac{\sum_{i=1}^{N} x_{ij} \times \nu_{mem}}{P_{mem}}
\]

\[
U_{BW}^j = \frac{\sum_{i=1}^{N} x_{ij} \times \nu_{BW}}{P_{BW}}
\]

where \( \nu_{cpu}, \nu_{mem}, \) and \( \nu_{BW} \) signify the CPU, reminiscence, and system bandwidth anxieties of \( VM_j \) correspondingly. \( P_{cpu}, P_{mem}, \) and \( P_{BW} \) signify the CPU, reminiscence, and network bandwidth dimensions of \( PH_j \) correspondingly. \( x_{ij} \) is a binary adjustable representing whether \( VM_j \) is allocated to \( PH_j \) or not.

5. Performance Validation

This unit deals with the presentation analysis of the EAVMP-CSSA method with other prevailing methods in terms of different evaluation parameters. Table 1 and Figure 3 investigate the power consumption examination of the EAVMP-CSSA method with other methods under varying VMs. The experimental result highlights that the EAVMP-CSSA technique has attained a minimal power consumption over the other methods under distinct VMs.

For instance, with 54 VMs, the EAVMP-CSSA technique has attained a reduced power consumption of 4817 W whereas the AP-ACO, ACO, FFD, and Random techniques have achieved an increased power consumption of 5283 W, 5399 W, 5690 W, and 6853 W, respectively. Moreover, with 108 VMs, the EAVMP-CSSA technique has resulted a lower power consumption of 6097 W whereas the AP-ACO, ACO, FFD, and Random techniques have attained a higher power consumption...
consumption of 6562 W, 7376 W, 7783 W, and 9121 W, respectively. Furthermore, with 162 VMs, the EAVMP-CSSA technique has attained a reduced power consumption of 6504 W whereas the AP-ACO, ACO, FFD, and Random techniques have achieved an increased power consumption of 7725 W, 9005 W, 9761 W, and 14472 W, respectively.

Table 2 and Figure 4 examine the communication cost examination of the EAVMP-CSSA method with other methods under varying VMs. The simulation values pointed out that the EAVMP-CSSA technique has gained the least communication cost over the other methods underneath variable VMs. For occurrence, with 54 VMs, the EAVMP-CSSA method has an effective outcome with the least communication cost of 41.24 W whereas the AP-ACO, ACO, FFD, and Random techniques have accomplished a higher communication cost of 45.83 W, 52.32 W, 56.91 W, and 64.55 W, respectively.

Eventually, with 108 VMs, the EAVMP-CSSA technique has gained a reduced communication cost of 59.58 W whereas the AP-ACO, ACO, FFD, and Random techniques have resulted an increased communication cost of 65.32 W, 71.43 W, 80.22 W, and 89.01 W, respectively. Meanwhile, with 162 VMs, the EAVMP-CSSA technique has demonstrated better performance with the minimal communication cost of 73.72 W whereas the AP-ACO, ACO, FFD, and Random techniques have accomplished a maximum communication cost of 81.36 W, 92.44 W, 96.67 W, and 105.79 W, respectively.

Table 3 and Figure 5 assess the time examination of the EAVMP-CSSA method with other methods under varying bandwidth. The found simulation result depicts that the EAVMP-CSSA method has occasioned with the larger presentation with minimum time over the other approaches under varying bandwidth. For instance, under 100 Mbps bandwidth, the EAVMP-CSSA technique has portrayed effectual performance with the lesser time of 1.62 s whereas the AP-ACO, ACO, FFD, and Random techniques have depicted a higher time of 2.77 s, 2.98 s, 3.18 s, and 3.78 s, respectively.

Additionally, with 500 Mbps bandwidth, the EAVMP-CSSA technique has demonstrated with the lesser time of 1.12 s whereas the AP-ACO, ACO, FFD, and Random techniques have taken an increased time of 1.45 s, 1.96 s, 1.85, and 3.40 s, respectively. At last, with 900 Mbps bandwidth, the EAVMP-CSSA technique has gained an optimal outcome with the minimal time of 0.09 s whereas the AP-ACO, ACO, FFD, and Random techniques have accomplished with the maximum time of 0.11 s, 0.54 s, 0.63 s, and 3.03 s, respectively.
Finally, a service rate examination of the EAVMP-CSSA method takes residence beneath varying number of VMs in Table 4 and Figure 6. The figure proves that the EAVMP-CSSA system has outperformed over the other methods with the maximum service rate for all the VMs. For instance, with 54 VMs, the EAVMP-CSSA technique attains an effective outcome with the maximum service rate of 98.12% whereas the Random, FFD, ACO, and AP-ACO techniques have gained a minimum service rate of 74.40%, 78.80%, 90.70%, and 96.31%, respectively. In the same way, with the presence of 162 VMs, the EAVMP-CSSA technique has illustrated proficient performance with an increased service rate of 59.20% whereas the Random, FFD, ACO, and AP-ACO techniques have gained a minimum service rate of 38.00%, 42.10%, 51.20%, and 54.80%, respectively. From the above benches and statistics, it is obvious that the EAVMP-CSSA method is originate to be an effective instrument for VMP in CDCs.

6. Conclusion

This paper has designed a novel EAVMP-CSSA method to achieve liveliness competence in CDCs. The EAVMP-CSSA method is mainly based on the design of CSSA with the integration of chaotic maps and conventional SSA. In addition, the EAVMP-CSSA technique derives an objective function to reduce energy utilization and resource wastage (e.g., CPU, RAM, and bandwidth). The proposed model has the ability to reduce the active server count by balancing the active servers that enables them to accommodate the upcoming VMP requests and eliminates the requirement of activating other servers. The performance of the EAVMP-CSSA technique is examined by the CloudSim tool and the results are investigated under different dimensions. The simulation results confirmed the betterment of the proposed EAVMP-CSSA technique over the recent state of art techniques. EAVMP-CSSA technique attains an effectual outcome with the maximum service rate of 98.12% whereas the Random, FFD, ACO, and AP-ACO techniques have gained a minimum service rate of 74.40%, 78.80%, 90.70%, and 96.31%, respectively. In the future, the design of EAVMP-CSSA technique can be extended to the design of task scheduling techniques to allocate resources in an optimal way. Besides, the presented technique can be employed to eradicate the overutilization of resources, which degrades the VM performance.

Data Availability

The manuscript contains all of the data.

Conflicts of Interest

The authors state that they do not have any conflicts of interest.

References


