International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/24 Volume 11, Issue 3, May – June (2024)

RESEARCH ARTICLE

SOAVMP: Multi-Objective Virtual Machine Placement in Cloud Computing Based on the Seagull Optimization Algorithm

Aristide Ndayikengurukiye
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
lemure.dede@gmail.com

Rim Doukha
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
doukharim@gmail.com

Eric Niyukuri
Faculty of Engineering Sciences, University of Burundi, Bujumbura, Burundi.
ericniyukuri@gmail.com

Eric Muheto
Morgan Stanley, Montréal, Canada.
eric.muheto.1@gmail.com

Abderrahmane Ez-zahout
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
abderrahmane.ezzahout@um5.ac.ma

Fouzia Omary
Intelligent Processing and Security of Systems Team, Faculty of Sciences, Mohammed V University, Rabat, Morocco.
omary@fsr.ac.ma

Received: 17 March 2024 / Revised: 28 May 2024 / Accepted: 09 June 2024 / Published: 30 June 2024

Abstract – Virtual machine placement (VMP) involves selecting the most appropriate physical machine (PM) to run a virtual machine (VM) in cloud data centers (CDCs). Unfortunately, current VMP methods only consider limited resources, resulting in load imbalance and unnecessary activation of certain PMs in the data center (DC). This paper proposes a new approach called Multi-Objective Seagull Optimization Algorithm Virtual Machine (MOSOAVMP) to address these issue s and enhance resource management in CDCs. The aim is to optimize resource utilization, minimize energy consumption, reduce SLA violations, and improve overall DC efficiency. The aim is to achieve an optimal deployment that will meet these different objectives while minimizing the costs associated with operating the CDCs. The results show the proposed MOSOAVMP's efficiency compared with existing algorithms for the different measurements considered. These experimental results show that MOSOAVMP reduces resource wastages, and energy consumption by 5.44%, improves average CPU usage by 14.84%, memory usage by 11.54%, average storage usage by 5.37%, and average bandwidth usage by 6.88%.

Index Terms – Cloud Computing, Seagull Optimization Algorithm, Metaheuristics Algorithm, SLA, Virtual Machine Placement, Data Center, Power Consumption.

1. INTRODUCTION

In recent years, cloud computing has become a leading IT model for providing and managing services over the Internet.

Its adoption is now ubiquitous and constantly growing, offering great scalability and various services. Cloud computing significantly impacts our daily life, particularly through social and sensor networks. The rise of smart devices has accelerated its adoption, leading to rapid growth in the number and size of cloud data centers. Additionally, it helps reduce network infrastructure costs.

Various techniques and technologies are used to meet user requirements and optimize cloud performance. These include migration, which consists of transferring a virtual machine (VM) from a physical server (PM) with insufficient resources to another with the necessary resources. What's more, virtualization is central to cloud computing. Virtualization has revolutionized IT operations in data centers (DCs), where virtual machines (VMs) are deployed on physical machines (PMs) to run users' applications [1]. Optimum placement of these VMs on physical hosts is crucial to guarantee good performance. This VMP problem is an optimization challenge whose objective is to determine the best VM allocation among the available PMs. Placement can be static or dynamic, depending on whether PMs can be replaced due to changes in the DC environment; such as variations in workload, resource availability, or hardware constraints. Migration of VMs may be necessary to balance the load, even though this may result in a breach of service level agreements (SLAs) [2]. However, it is important to avoid excessive resource utilization, as this can lead to performance degradation. The Cloud also makes it easier to store and analyze large volumes of data [3], which means that several challenges, such as safety [4], need to be considered. While cloud computing offers many advantages, it also presents several challenges. One of the most important is to reduce energy consumption in cloud-based data centers. In addition, optimizing the use of different cloud resources is also crucial. To overcome these challenges, various metrics are taken into account and analyzed.

It is therefore essential to find an optimal solution for VMP that meets the above-mentioned objectives. This implies finding a balance between the different objectives, as improving one of them may lead to the deterioration of another. A multi-objective optimization approach is therefore needed to effectively solve this VMP problem.

This article presents a new algorithm, designed to address the challenge of optimizing the placement of VMs on PMs to improve IT resource management. The proposed algorithm has three main objectives. The first objective focuses on reducing energy consumption in CDCs by minimizing the number of active physical servers. The second objective is to reduce the waste of various cloud resources while optimizing the VMP. Finally, the algorithm aims to minimize and reduce the SLAs among active PMs in CDCs. In addition, the algorithm considers other factors such as CPU usage, memory, bandwidth, storage space, as well as the number of active machines and migrations.

The use of this solution significantly reduced the number of migrations to CDCs. The proposed MOSOAVMP algorithm was compared with other commonly used algorithms such as DMOSCA-SSA [4], MOILP [5], PIAS [6], MGGAVP [7], and MBFD [8]. As demonstrated earlier in the summary, the proposed approach demonstrated substantial improvements over the aforementioned methods. The metrics used for this comparison include, among others, energy consumption, utilization of various resources (CPU, RAM, storage, and network), migrations, as well as compliance with service level agreements (SLAs).

The remainder of this article is organized as follows: The second section reviews the various algorithms, instead of works. Section 3 gives a general introduction to the various mathematical concepts of the Seagull Optimization Algorithm (SOA), and is followed by the mathematical formulation of the multi-objective optimization applied to VMP, and the description of the proposed algorithm is found in section 4. Section 5 presents the results of the performance evaluation of the proposed method in comparison to other commonly used algorithms. Finally, section 6 provides conclusions and outlines prospects for future work.

2. RELATED WORK

As mentioned above, VMP aims to ensure good cloud performance and optimal resource management. There are many algorithms available to solve this problem, and in this section, we review some of the work that addresses the VMP problem.

Lu et al. [9] proposed an improved genetic algorithm (I-GA) to address the VMP problem, aiming to optimize the availability and energy consumption of CDCs. A combination of virtual hierarchy architecture and GA was utilized to achieve a near-optimal solution, to improve energy consumption and resource availability. A key step in their approach is the generation of the initial I-GA population, which is achieved using finite element analysis in the background. CloudSim is employed for simulating the experiments. The results demonstrate a significant improvement in energy management and high availability in the DCs. Their model presents better results based on the benchmark results of the different state-of-the-art methods used for the VMP optimization problem. The authors efficiently optimize the VMP by utilizing the I-GA approach, contributing to better resource management and reduced energy consumption in CDCs.

Caviglione et al [10] proposed a multi-objective approach to determine the best techniques for VMP. It takes into account the impact of hardware failures, DC power consumption, and user satisfaction in terms of performance. A deep

RESEARCH ARTICLE

reinforcement framework facilitates the selection of the best heuristic for the placement of VMs. Their results demonstrate that the method exceeds the performance of current state-of-the-art heuristics for both real and synthetic workloads [10].

Alharbi et al [11] introduced a method combining the assignment of different applications and the VMP of a DC to optimize the energy efficiency of enterprise distribution centers. They formulated the problem related to energy optimization of enterprise DCs as an Int2LBP (Integrated Two-Layer Bin Packing) problem. This approach was considered as an initial solution and was further developed to optimize the energy efficiency of corporate distribution centers. In terms of energy efficiency, the various simulations carried out on DC prove the performance of the Int2LBP_FFD algorithm compared with the Consec2LBP_FFD algorithm. In addition, the Int2LBP_ACS algorithm outperforms the Int2LBP_FFD in efficient energy management. The Int2LBP_FFD and Int2LBP_ACS algorithms are well suited to managing different applications and VMs on DCs for large enterprises.

Ghetas et al. [12] introduced a new method called MBO-VM based on the VMP Monarch Butterfly Optimization (MBO) algorithm. This method aims to maximize packaging performance and reduce PMs. The CloudSim simulator was utilized to implement and assess the performance of this approach. This tool also enabled them to test real and synthetic workloads in the cloud. The results obtained from the simulations demonstrate that MBO-VM outperforms known VMP techniques in terms of performance. By using MBO-VM, it is possible to more effectively reduce the number of active hosts while maximizing packaging efficiency. This approach thus offers an optimal solution for VMP, improving the overall efficiency of CDCs.

To solve the VMP optimization problem, a new multi-objective ILP algorithm has been presented by Regaieg et al [5], considering that at cloud service providers (CSPs) the environment is made up of VMs of homogeneous and heterogeneous types. This model aims to reduce resource wastage while maximizing the number of VMs that are hosted on physical servers, thereby reducing the number of PMs used. Due to the diversity of the different types of VMs available in a heterogeneous cloud environment, the test results showed the effectiveness of achieving the above objectives. These results have the advantage of reducing the costs associated with operating DCs while maintaining a high level of Quality of Service (QoS).

Rashida et al. [13] presented a VMP algorithm called MGGAVP, which considers correlation and addresses the VMP problems. This algorithm is based on escalation algorithm hybridization and genetic clustering and is extended to operate in a multi-cloud environment. After simulation, the results reveal the net performance offered by their algorithm to those of the other benchmark algorithms. It achieves energy savings of 51.93% and reduces energy costs by 70.41%.

Rahimi Zadeh et al. [6] proposed a new scheduling scheme called PIAS (Profit-aware Interference-aware Scheduling) for the efficient consolidation of VMs in the Infrastructure-as-a-Service (IaaS) model. The PIAS scheme takes into account several factors such as profit, energy costs, operational interference, resource utilization, and SLAs. To minimize workload execution costs and increase vendor profit, the optimization problem was modeled in the form of stochastic dynamic programming, mimicking the operational behavior of VMs. Simulations based on real workloads show that PIAS outperforms competing approaches on average, with improvements of at least 40% for profit, 29% for energy efficiency, and 35% for service downtime.

To improve quality of service (QoS) and efficient traffic management in Internet of Things (IoT) networks, a meta-heuristic based on enhanced seagull optimization is proposed by Gharehpasha et al. [14]. This approach enables better management of packet forwarding and a significant improvement in QoS in terms of delay. The performance of this method has been evaluated and compared with previous methods, demonstrating its accuracy, efficiency, and superiority.

Nabavi et al. [15] presented a multi-objective approach to VMP in edge-cloud DCs. To optimize network traffic and traffic power, an SOA-based model was presented. The strategy focuses on trying to reduce network traffic between PMs while consolidating VM communications on the same PMs. This reduces data transfer across the network and lowers PMs power consumption. The authors conducted simulations using CloudSim and tested the proposed approach on two network topologies, VL2, and three-tier. The results highlight the proposed method's effectiveness in significantly decreasing energy consumption and network traffic in edge-cloud environments. Test results for the proposed algorithm show a clear reduction in energy efficiency of 5.5%, a remarkable 70% reduction in network traffic, and an 80% reduction in the energy consumption of network components.

Table 1 provides a summary of these various state-of-the-art works. It presents the methodology used, the objectives, the weaknesses, the simulator used, and the proposed method.

The choice of the MOSOAVMP algorithm to optimize the placement VMs in the cloud is based on the inherent qualities of the SOA algorithm. These include its simplicity, ease of implementation, and ability to converge rapidly on optimal solutions, even with many iterations. This testifies to its robustness and ability to solve complex problems such as VM placement. The analysis presented in Table 1 reveals a thorough evaluation of the metrics used in previous work, enabling us to discern which metrics are paramount and which are often overlooked. This analysis enabled us to select a wide range of metrics relevant to our study, ensuring a comprehensive and rigorous assessment of cloud performance.

Table 1 Summary of the Work Reviewed

Algorithm	Vulnerability	Objectives	Methodology Applied	Simulator
I-GA [9]	Very long runtimes and certain metrics such as storage and bandwidth are not taken into use.	Ensure availability and optimize the energy consumption of a DC	Proposal of an I-GA algorithm to optimize VMP problems.	CloudSim
DLR-VMP [10]	Resource wastage	VM placement while ensuring Quality of Experience (QoE), by reducing the power used.	Placement strategies are determined through a deep reinforcement learning framework that identifies the optimal placement heuristic for each VM within the workload.	Python
ACS, Int2LBP [11]	Runtime execution, applications are allocated dynamically, dynamic VMP, VM consolidation, and VM migration	Implementation of the Int2LBP_FFD algorithm and the Int2LBP_ACS algorithm, based on the initial results of the first algorithm, to optimize the energy efficiency of corporate distribution centers.	To solve the problem of energy consumption in an enterprise DC, the authors call it Int2LBP	Amazone EC2 T2
MBO-VM [12]	Storage, Bandwidth, and SLA	Minimize the number of active physical hosts in the DC to optimize energy consumption and lower maintenance costs.	Server consolidation and resource utilization	CloudSim
MOILP [5]	The effect of workload on efficient energy consumption is not taken into account.	Minimize DC operating costs to increase customer satisfaction.	Optimizing hosted VMs, limiting the waste of different resources, and reducing the number of active physical servers to reduce energy consumption in DCs.	Amazone EC2, Julia Language (JL)
MGGAVP [13]	QoS, SLA	Hybridization of metaheuristic escalation algorithms and genetic clustering for optimization in a multi-cloud environment	Optimization of the VMP problem in a heterogeneous environment.	MATLAB R2015
PIAS [6]	Lack of peak period prediction to increase or reduce DC capacity during peak or idle periods. To avoid saturation, VM resource utilization should remain below.	Cost of energy efficiency, operational VM interference, reduced resource utilization, and SLA optimization.	VM behavior is modeled using stochastic dynamic programming to minimize workload execution costs and maximize provider profit.	CloudSim
ISOA [14]	Many of the metrics (e.g. CPU, RAM, storage, ...) to assess the efficiency of the cloud have not been taken into account.	improve QoS and efficient traffic management in Internet of Things (IoT) networks.	Use SOA to minimize lead times, average implementation times, virtual server computing costs, and network costs.	MATLAB R2017

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/24 Volume 11, Issue 3, May – June (2024)

SOAVMP [15]	Little use of test resources, SLA, and resources wastage	Traffic, power, and energy	Reduced energy consumption thanks to consolidation technology and reduced network traffic between PMs by concentrating as many VMs as possible on a single PM.	CloudSim
Our method	Security and task scheduling	Energy consumption, SLA, resource utilization(CPU, RAM, Storage, and Bandwidth), Migration, and resource wastage.	MOSOAVMP optimizes virtual machine placement	CloudSim

3. SEAGULL VIRTUAL MACHINE PLACEMENT OPTIMIZATION ALGORITHM

3.1. Bio-Inspired Paradigm

Optimization in general is a field that has been developing steadily over the last few years, with new bio-inspired algorithms [16], [17], [18] as well as hybrid algorithms [19], [20]. Inspired by the natural migration and attack patterns of seagulls in the wild, SOA is a metaheuristic that was proposed by Dhiman and Kumar in 2017[21]. This algorithm is population-based and uses migratory movements to search for abundant food sources. Their positions are updated according to the best position found, while fascination and attack strategies are used to attract prey. The algorithm is divided into diversification and intensification, the two main phases, and can switch from one phase to the other depending on the position of the prey. Seagulls can continuously modify their angle of attack during migration [22].

The migratory behavior of seagulls is characterized by the following features [23]:

- Seagulls move in groups during migration, adopting a collective movement strategy to avoid collisions between them;
- Within the group, seagulls can adjust their trajectory by moving in the direction that ensures the best survival for the oldest seagull. In other words, they may follow the direction chosen by the most experienced seagull to maximize their chances of survival.

This noisy bird lives, feeds, and sleeps in colonies. Its food consists of insects, their larvae, earthworms, small crustaceans, mollusks, and small fish caught in flight. As shown in Figure 1, during the attack they perform a spiral movement.

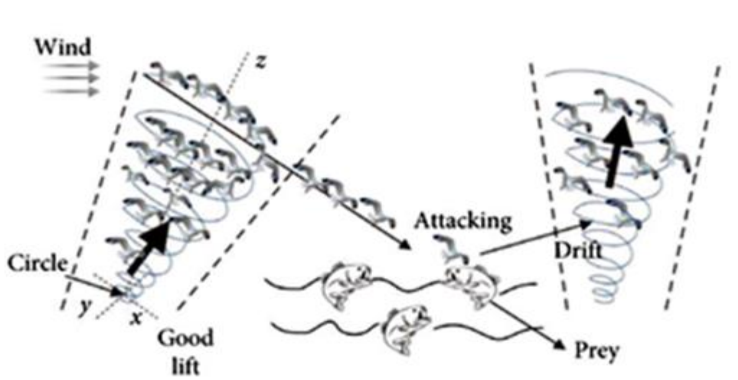

Figure 1 Migration and Attack Phases of Seagull Behavior [21]

3.2. Mathematical Formulation

This algorithm simulates the movement of a group of seagulls from one location to another during migration. During this migration phase, three conditions must be met by each of the seagulls:

3.2.1. Migration phase (or Exploration phase)

Collision Avoidance: To prevent collisions with other seagulls when computing the new position of the search agent, a variable named A is introduced. [24].

$$\overrightarrow{C_s} = A \times \overrightarrow{P_s}(x) \qquad (1)$$

In Equation (1), $\overrightarrow{C_s}$ is a representation of the search agent's position that avoids collisions between neighbors, P_s represents its current position, x designates the current iteration, and A characterizes the agent's various movements within the search space. Figure 2 shows how search agents avoid colliding with each other. These variables are applied to compute the optimal position for the search agent while avoiding collisions with other agents[24], [25]. This is shown in Equations (2) and (3).

$$A = f_c - \left(x \times \left(\frac{f_c}{Max_{it}} \right) \right) \tag{2}$$

$$x = 0,1,2, \ldots, Max_{it} \tag{3}$$

In Equation (2), variable A reproduces the motion behavior of the search agent. The parameter f_c controls this variable and decreases nearly from f_c to 0. The maximum number of iterations used is represented by Max_{it}. These elements are essential for regulating the agent's movement and defining the limits of the optimization process. Figure 3 shows how the different agents move and search for the best neighbor.

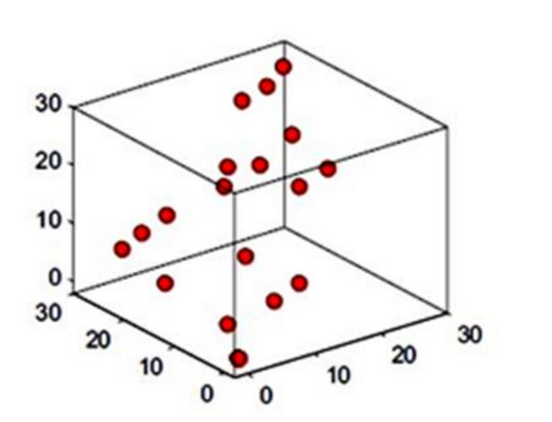

Figure 2 Avoiding Search Agent Collisions

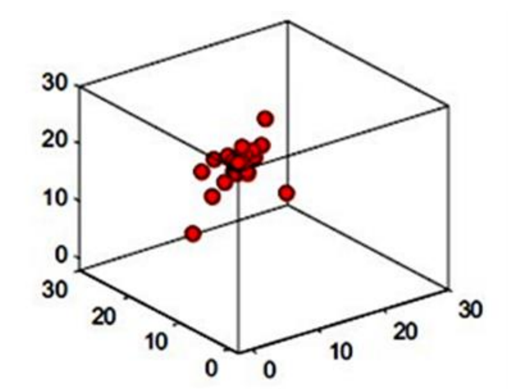

Figure 3 Moving Agents and Finding the Best Neighbor

Seagulls head for the best-performing seagull in their group, ensuring cohesion. This is represented in the mathematical equation (4)[26], [27]:

$$\overrightarrow{M_s} = B \times \left(\overrightarrow{P_{bs}}(x) - \overrightarrow{P_s}(x) \right) \tag{4}$$

Where $\overrightarrow{M_s}$ represents the position of the $\overrightarrow{P_s}$ of gulls in relation to the P_{bs} of the best-performing seagulls. Control parameter B, as indicated by reference [28], plays a pivotal role in establishing equilibrium between diversification and intensification. It is represented mathematically in Equation (5)[27],[29] where Ran denotes a random number within the range of [0; 1][28]:

$$B = 2 \times A^2 \times Ran \tag{5}$$

Approaching the Optimal Seagull: Seagulls actively seek out food sources during their migration, moving in the direction that offers the best opportunities for survival. Figure 4 illustrates the updating of the search agent's position relative to the position of the best search agent.

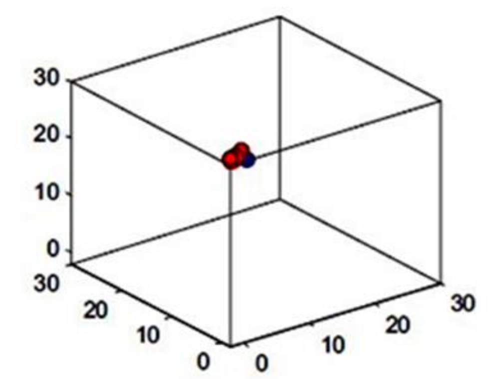

Figure 4 Approaching the Optimal Search Agent

In this case, updates of other seagull's positions rely on the position of the best seagull. The Equation (6) below represents this update [30]:

$$\overrightarrow{D_s} = \left| \overrightarrow{C_s} + \overrightarrow{M_s} \right| \tag{6}$$

3.2.2. Attack Phase (or Exploitation Phase)

In the exploitation phase, the seagulls utilize their previous knowledge and experience. They can constantly adjust their angle of attack and speed. To maintain altitude, they use their wings and body weight.

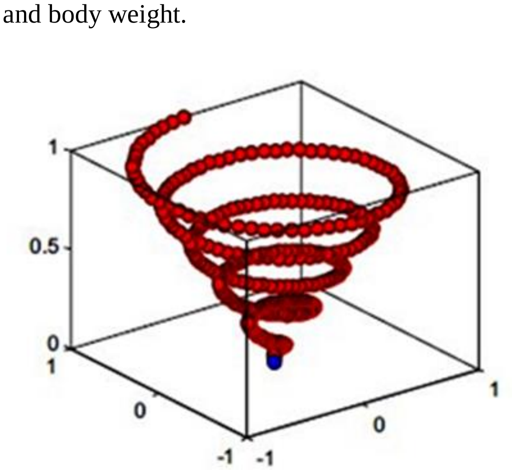

Figure 5 Natural Behavior of Seagull Attack

RESEARCH ARTICLE

As illustrated in Figure 5, the seagulls engage in a spiral motion while attacking their prey in the air. The Equations (7)-(10) below, calculate the three dimensions (x, y, z) of this behavior [31], [32], [33]:

$$x = r \times \cos(\theta) \qquad (7)$$
$$y = r \times \sin(\theta) \qquad (8)$$
$$z = r \times \theta \qquad (9)$$
$$r = v \times e^{\theta u} \qquad (10)$$

Updating the search agent's position within the spiral involves the utilization of various parameters. The radius of the spiral is represented by r, while θ is a random element between$[0 < \theta < 2\pi]$[34]. The parameters and u influence the shape of the spiral. Using the natural logarithm of base e, the following Equation (11) calculates the updated position of the search agent [35], [36]:

$$\overrightarrow{P_s}(x) = \overrightarrow{(D_s} \times x \times y \times z) + \overrightarrow{P_{bs}}(x) \qquad (11)$$

The $\overrightarrow{P_s}(x)$ variable in this equation plays an essential role in maintaining the best available option and taking into account the current situation of the other search agents. It maintains a reference to the best solution throughout the process.

4. MATHEMATICAL FORMULATION AND MOSOAVMP FOR VIRTUAL MACHINE PLACEMENT

4.1. Problem Formulation

The optimal placement of VMs in CDCs is pivotal in cloud computing. The primary aim is to minimize energy consumption, resource wastage, SLA violations, and maximize DC efficiency.

Cloud computing presents complex challenges when it comes to VMP. The issue of VMP in a CDC is unpredictable and follows no consistent pattern. The complexity of this problem can be illustrated by the fact that the maximum number of VM mappings on PMs is equal to m^n , where n is the number of VMs and m is the number of PMs. With this context, we determine the first objective, which is to minimize the energy consumption of the PMs in the CDC. The various mathematical symbols used in the mathematical formulation are described in Table 2. Recent research has shown a linear correlation between CPU usage and server energy consumption in CDC. A linear equation can be used to precisely calculate energy consumption in the context of cloud computing. This relationship shows that when CPU utilization is increased, host power demand increases proportionally [37], [38], [39]:

$$P_i^{power} = \begin{cases} \left(P_i^{busy} - P_i^{iddle}\right) \times U_i^{cpu} + P_i^{iddle} & if\ U_i^{cpu} > 0 \\ 0, & otherwise \end{cases} \qquad (12)$$

Equation (12) is used to determine the energy consumption of a PM in a DC. These different variables P_i^{power} , P_i^{busy} , P_i^{iddle} and U_i^{cpu} represent, respectively, the amount of energy consumed when the machine is fully loaded, idle, and inactive, and CPU utilization in MIPS. According to this equation, power consumption is directly proportional to CPU utilization. As a result, an increase in CPU will lead to an increase in power consumption in the DCs.

Energy consumption in CDCs is directly linked to CPU power consumption. The increase in the CPU usage of the PMs will directly influence the increase in the energy consumption of the CDCs. Consequently, the overall energy consumption in the CDCs can be calculated based on the following Equation (13):

$$\sum_{i=1}^{m} P_i^{power} = \sum_{i=1}^{m} b_i \times ((P_i^{busy} - P_i^{idle}) \times \sum_{v=1}^{m} a_{jp}.C_j + P_i^{idle} \qquad (13)$$

Table 2 Various Mathematical Symbols and Descriptions

Symbol	Descriptions
P	Group of PMs
V	Group of VMs
α	F1 balance coefficient
β	F2 balance coefficient
δ	F3 balance coefficient
a, b	Binary variables
P_i^{iddle}	Power consumption in inactivity mode
P_i^{power}	Power consumption of PM i
P_i^{busy}	Energy consumption of PMs during periods of activity
U_i^{cpu}	The normalized CPU utilization of PM i
U_i^{memo}	The normalized RAM utilization of PM i
$R_i^{wastage}$	Inefficient use of PM resources
NR_i^{cpu}	Normalized remaining CPU of PM
NR_i^{memo}	Normalized remaining memory of PM
TU^{CPU}	Global CPU usage PMs in operation
TU^{memo}	Global memory usage PMs in operation
TU^{sto}	Global storage usage PMs in operation

RESEARCH ARTICLE

TU^B	Global bandwidth usage PMs in operation
T_i^{cpu}	Peak CPU usage of PMs
T_i^{memo}	Peak memory usage of PMs
T_i^{sto}	Peak storage usage of PMs
T_i^{band}	Peak bandwidth usage of PMs
C_j	CPU request of VM j in MIPS
M_j	Memory request of VM j in MB
H_j	Storage request of VM j in GB
B_j	Bandwidth request of VM j in Mbps

Preventing wasted resources is a crucial aspect of placing VMs in CDCs. Each server has hardware resources that must be optimally utilized to host VMs. Efficient management of unused server resources is essential. Equation (14) quantifies the waste of resources [40].

$$R_i^{wastage} = \frac{|\, NR_i^{cpu} - NR_i^{memo} \,|}{|\, U_i^{cpu} + U_i^{memo} \,|} + \varepsilon \qquad (14)$$

Equation (14) is used to quantify the remaining unused resources on a PM p in a CDC. NR_i^{cpu}p represents the amount of unused CPU, while NR_i^{memo}p represents the amount of memory unused by VMs on that same PM i. The variables U_i^{cpu} and U_i^{memo} represent CPU and memory usage, respectively, by VMs on the PM i in the CDC. Equation (15) represents the total quantity of the various resources consumed in (CDC).

$$\sum_{i=1}^{m} R_i^{wastage} = \sum_{p=1}^{m} \left[a_i \times \frac{\left|(T_i^{cpu} - \sum_{j=1}^{n}(b_{ji}.C_j)) - (T_i^{memo} - \sum_{j=1}^{n}(b_{ji}.B_j))\right| + \varepsilon}{(\sum_{j=1}^{n}\left(b_{ji}.C_j\right) + (\sum_{j=1}^{n}(b_{ji}.M_j)} \right] \qquad (15)$$

When the values a_i and b_{vi} are equal, this indicates that the PM P is active and hosting the VM v.

To guarantee a minimum level of service between the cloud service provider and the customer, an SLA contract must be respected. Equation (16) represents this SLA:

$$SLA = \frac{1}{1 + e^{U_{cpu} - 0.9}} \quad (16)$$

The objectives of placing VMs on PMs in CDCs are defined from the previous equations, and include the following:

$$Min\ F(x) = \alpha F_1(x) + \beta F_2(x) + \delta F_3(x) \qquad (17)$$

The objective function used in Equation (17) takes into account several objectives in the placement of VMs. The coefficients α, β and δ keep these objectives in balance. The first objective, F1, concerns the energy consumption of the PMs. The second objective, F2, measures resource wastage. Finally, the third objective, F3, takes into account SLA compliance. The values of these functions can be determined through the following Equations (18)-(20):

$$Min \sum_{i=1}^{m} P_i^{power} \qquad (18)$$

$$Min \sum_{i=1}^{m} R_i^{wastage} \qquad (19)$$

$$Min \sum_{i=1}^{m} SLA \qquad (20)$$

When optimally placing VMs on PMs in a CDC, several constraints must be taken into account, including:

$$\sum_{i=1}^{m} b_{ji} = 1 \qquad (21)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, C_j \le T_i^{cpu} .\, a_i TU^B \qquad (22)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, M_j \le T_i^{memo} .\, a_i \qquad (23)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, H_j \le T_i^{sto} .\, a_i \qquad (24)$$

$$\sum_{j=1}^{m} b_{ji} = b_{ij} \,.\, B_j \le T_i^{band} .\, a_i \qquad (25)$$

Each VM is assigned to a single PM, as shown in equation (21). Equations (22), (23), (24) and (25) ensure that the cumulative of resources (CPU, memory, storage space, and bandwidth) demanded by the various VMs on a PM cannot exceed the capacity of that PM.

4.2. Proposed Algorithm

In CDCs, a collection of homogeneous or non-homogeneous VMs utilize the hardware resources of the PMs to run their different services. Hosting these VMs is the responsibility of the PMs in these centers. More hardware resources are needed to meet the growing request for cloud services, and this is driving up costs. Optimizing VMP will maximize the use of available resources while enabling each PM to host a significant number of VMs. This approach avoids wasting the various resources available in the cloud and also makes the most of the processing power, memory, and storage systems of the PMs. This guarantees the performance of a cloud-based DC.

This study presents a new solution based on MOSOAVMP, a discrete multi-objective approach to optimal VMP. The main objective is to optimize resource management, minimize energy consumption, and reduce the SLA in the CDC. Algorithm 1 shows the solution proposed in this work.

Input: Initializing the agent population

Output: The best solution for optimizing VMP

Begin

1. Initialize SOA Parameters
2. Set the population size of the Seagulls
3. Compare the cost function of each agent
4. Determine the maximum number of iterations
5. Initialize the seagull position randomly

While t < maxiter do

For each search agent do

Apply mutation and crossover on the solutions as migration and seagull behavior

Calculate objectives values for all search agents

End for

Update seagull position

For each i=1 to number of seagulls do

Compute the costs involved with using all search agents

Generate the most effective solutions using the latest search agents

End for

Consider the most efficient solution.

End While

Return the best solution

Algorithm 1 MOSOAVMP

The first phase of all metaheuristic algorithms is the random phase called initialization, where random solutions are generated within the search space. This phase significantly influences the algorithm's efficiency and final results. To achieve an optimal solution, a specific number of agents is defined. In this version of AOS, which is proposed in Algorithm 1, the various steps are as follows:

Step 1: This is the crucial step of initializing the parameters taken into account in this algorithm. It involves mapping VMP in the CDC to the seagull and resetting the VM position. The maximum number of iterations, the number of agents, and the dimensions of the space are taken into account during initialization;

Step 2: This is the first phase in the search for the optimal solution. This solution is obtained as a function of resource constraints, using the equation representing the objective function. The solution with the smallest value is then considered to be the optimal solution;

Step 3: This step involves updating the seagull's position;

Step 4: The search process is interrupted if the maximum number of iterations is reached. At this point, the position of the leading agent is considered. If the maximum iterations are not yet reached, the process returns to step 2.

5. RESULTS AND DISCUSSIONS

5.1. Performance and Evaluation

This section presents the configuration used, the various performance measurements, as well as the different simulation results of the new algorithm(MOSOAVMP), evaluated and compared against various existing algorithms dealing with the VMP problem such as DMOSCA-SSA MOILP, MBO-VM, PIAS, MGGAVP, and MBFD.

The main objective is to perform the optimal placement of VMs on PMs in a Cloud environment to limit resource wastage, energy consumption costs, and SLA, and maximize total performance. In this study, we took into account several key evaluation metrics such as resource wastage, energy consumption, and utilization of different resources (CPU, memory, storage, and bandwidth). Added to this are the number of active machines and migration.

The experimental study of the MOSOAVMP algorithm was carried out using the CloudSim simulator version 5.0[41]. The CloudSim simulator supports the modeling and simulation of a cloud-based DC [42]. It was chosen for its flexibility in implementing many IaaS functionalities such as energy management, and resource management, as well as evaluating new applications before implementation in a real environment.

The experiments were carried out on a computer equipped with an AMD Ryzen 5800U processor clocked at 4.4 GHz with 8 cores and logical processors equal to 16, plus 16 GB of 3200MHz RAM, running the Windows 11 operating system.

To conduct simulations of the heterogeneous Cloud environment, a single DC is established, comprising 1800 PMs with two configurations, as outlined in Table 3.

This is done for a set ranging from 200 to 2,000 VMs, with an interval of 200 VMs for each iteration. The specifications of the VMs are detailed in Table 3.

International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/24 Volume 11, Issue 3, May – June (2024)

RESEARCH ARTICLE

Table 3 Different Test Configurations

		Host name	Parameter and value
Host Type	PMs	HP ProLiant ML110 G3	MIPS: 3000 RAM: 4GB Cores: 2 Storage: 1024 GB Bandwidth: 3Gbps
		HP ProLiant G4	MIPS: 3720 RAM: 4GB Cores: 2 Storage: 1024 GB Bandwidth: 3Gbps
	VMs	-	MIPS: 2000 RAM: 1024MB Cores: 1 Storage: 2.5GB Bandwidth: 512MB

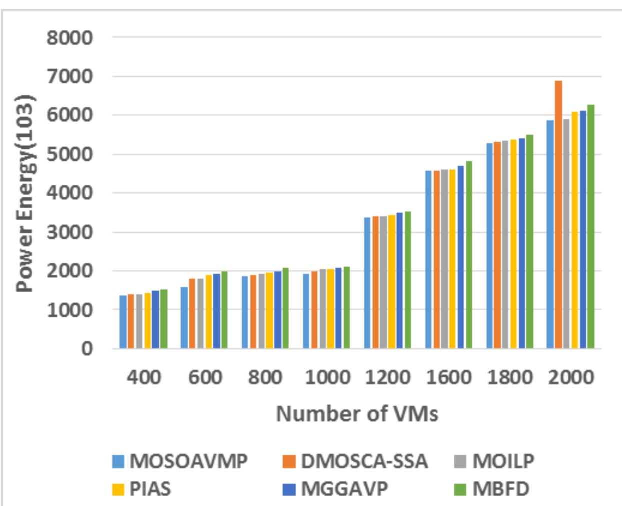

Figure 6 Power Energy Consumption

Figure 6 shows the energy consumption results for the algorithms used in the tests. Energy consumption increases with the number of VMs active in the CDC. In all cases, MOSOAVMP shows a clear improvement in energy consumption over the other algorithms compared. Considering the basic configuration of 400 VMs, the proposed MOSOAVMP model shows a low power consumption of 5.28% on average, compared with the state-of-the-art DMOSCA-SSA, MOILP, PIAS, and MGGAVP algorithms. For the maximum configuration of 200 VMs, MOSOAVMP demonstrates a 6.18% improvement over the other models evaluated. Overall, the proposed algorithm achieves a 5.44% reduction in energy consumption compared to other advanced algorithms.

Figure 7 shows VM migration comparisons. The proposed method shows the performance and stability of migration numbers. To migrate these VMs, we consider the resources available on the physical servers. In the case of state-of-the-art algorithms, VM correlation, and sudden resource changes are not taken into account, generating a large number of VMs that need to be migrated. The proposed algorithm limits VM migration while respecting the constraints since it is based on resource correlation. To achieve the best performance, the proposed algorithm minimizes the competition that can arise between VMs hosted on the same host by selecting the most appropriate VMs and PMs.

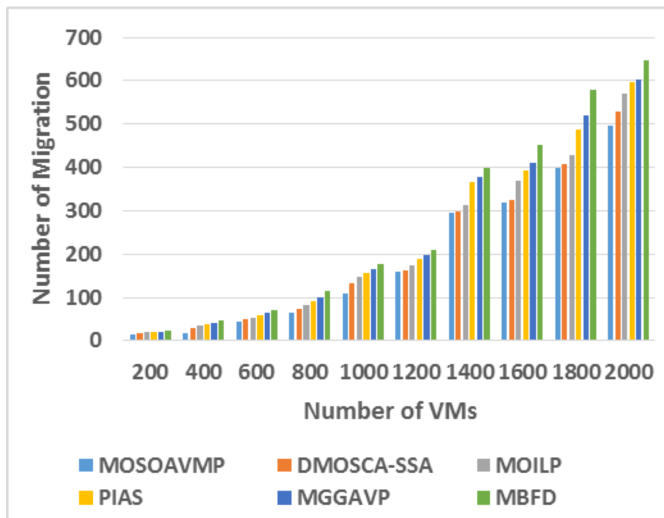

Figure 7 Number of Migration

The results in Figure 7 show the effectiveness of MOSOAVMP with the lowest migration costs, followed by DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD. In general, VM migration involves a considerable amount of data transfer. To obtain an optimal VMP, we reduce the number of migrations while placing VMs on the minimum possible PMs. In this way, the proposed algorithm reduces the number of active PMs and network links, while prioritizing the placement of VMs. In this way, the algorithm will reduce the number of active servers while prioritizing the placement of correlated VMs on the same server or neighboring servers.

Figure 8 shows the results for the number of active PMs in proportion to the number of VMs hosted on them. The test results show that the proposed MOSOAVMP algorithm

RESEARCH ARTICLE

requires fewer physical hosts to host VMs than the state-of-the-art algorithms. This improvement becomes significant as the number of VMs assigned to it increases. On average, MOSOAVMP improves by 7.92% over the compared algorithms. For example, let's compare MOSOAVMP with DMOSCA-SSA, the second best-performing algorithm. We see an improvement of 5.22% when using a configuration of 200VMs for both algorithms and an improvement of 1.22% in the case of 2000VMs. In the case of the MBFD algorithm, which presents the weakest results, MOSOAVMP shows a clear improvement of 19.92% for the 200VMs configuration and 8.10% for the 200% configuration. For all test configurations, MOSOAVMP performed well against DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD.

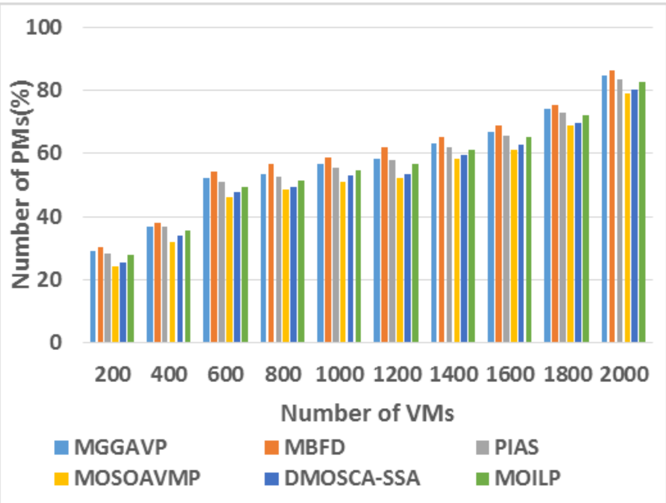

Figure 8 Number of Active PMs

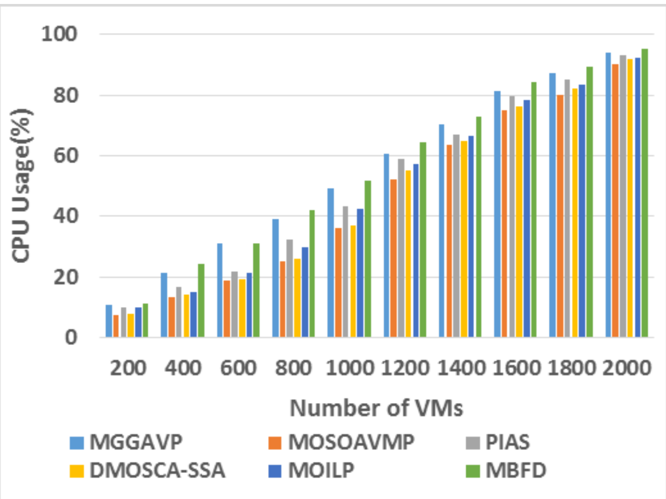

Figure 9 Average CPU Usage

Figure 9, Figure 10, Figure 11, and Figure 12 represent respectively the average use of PM resources such as CPU, memory, storage, and bandwidth as a function of active VMs in the DC. The results demonstrate the effectiveness of the proposed method in all situations. For CPU usage, MOSOAVMP shows an average improvement of 14.84% compared to state-of-the-art algorithms, followed by DMOSCA-SSA. For memory, MOSOAVMP shows an 11.54% improvement, a 5.37% improvement in storage space, and a 6.88% improvement in bandwidth.

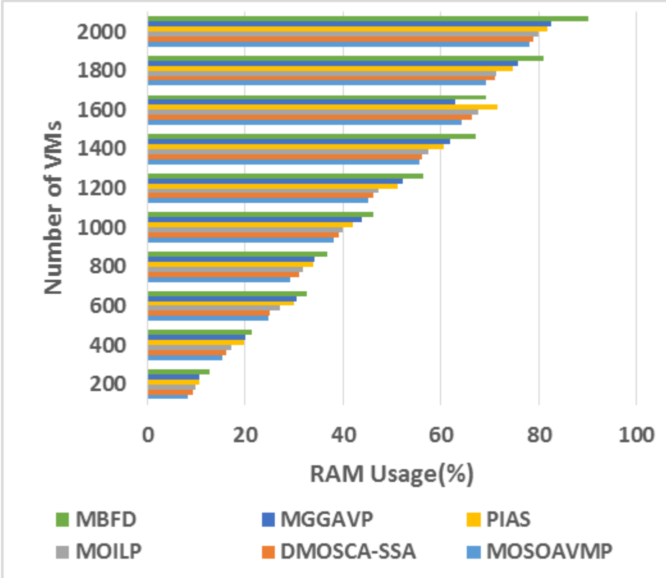

Figure 10 Average RAM Usage

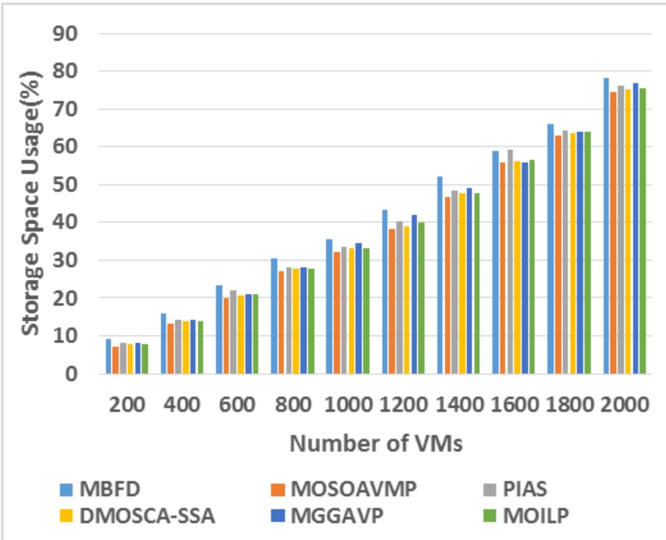

Figure 11 Average Storage Usage

Maintaining effective resource management in a cloud environment is a primary objective. Resource wastage significantly impacts cloud computing performance, as highlighted in the other metrics discussed earlier. According to the findings illustrated in Figure 13, the proposed algorithm effectively reduces resource wastage in the cloud when compared to existing algorithms.

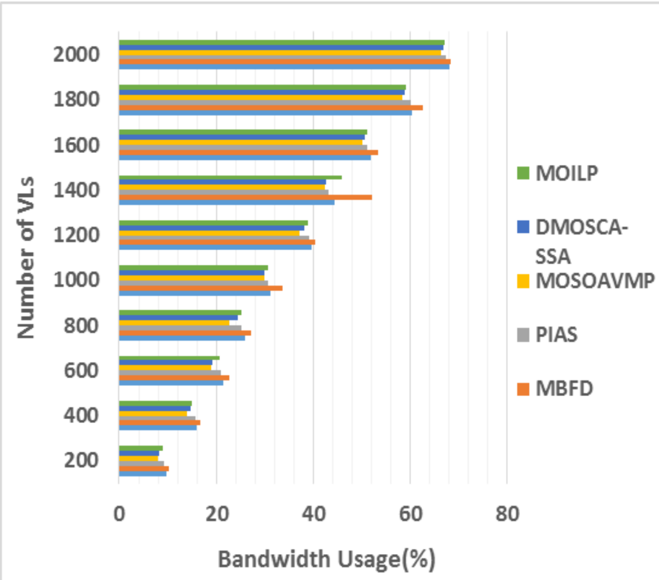

Figure 12 Average Bandwidth

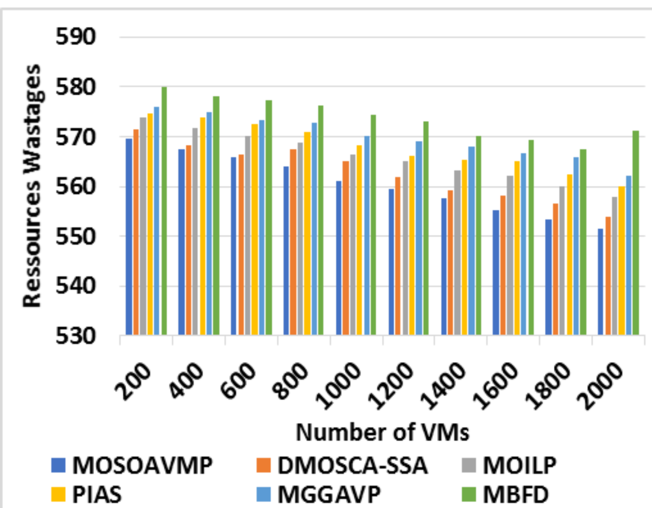

Figure 13 Resources Wastages

SLA reduction translates into a more reliable user experience and customer satisfaction. Figure 14 shows the effectiveness of the proposed MOSOAVMP algorithm when managing SLA in cloud computing, compared with DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD.

The various previous fluctuations observed with the existing algorithms have been reduced. As shown in Figure 14, the results demonstrate the superior performance of MOSOAVMP compared to state-of-the-art algorithms.

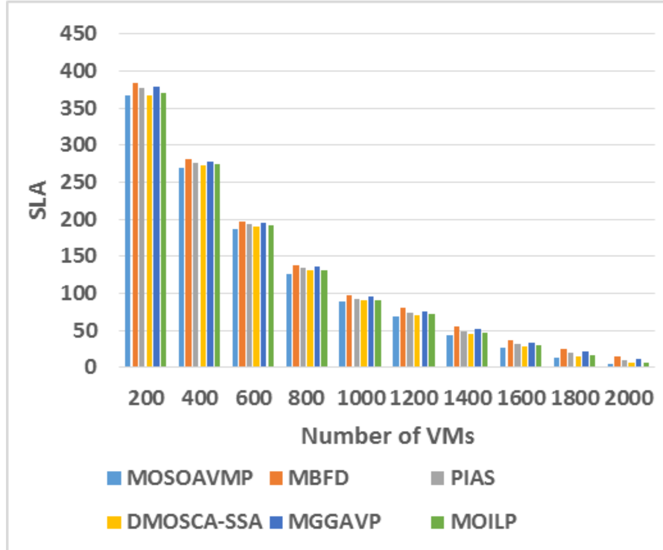

Figure 14 SLA

The convergence time for finding an optimal solution to various problems is crucial for metaheuristics. This includes the time elapsed during both the exploration and exploitation phases. To evaluate the performance of MOSOAVMP against state-of-the-art algorithms, the number of iterations is set to 100. Figure 15 illustrates the performance of the proposed method in comparison to other algorithms. As the number of iterations increases, the fitness of the DMOSCA-SSA, MOILP, PIAS, MGGAVP, and MBFD algorithms decreases. Under the same conditions, the MOSOAVMP algorithm demonstrates high efficiency. Increasing the number of iterations allows the proposed method to converge faster to the optimal solution than the other algorithms. Like any metaheuristic, it must balance between the local optimum value and the global optimum value throughout the search for the best solution. This balance determines MOSOAVMP's ability to effectively solve complex problems, particularly VMP problems.

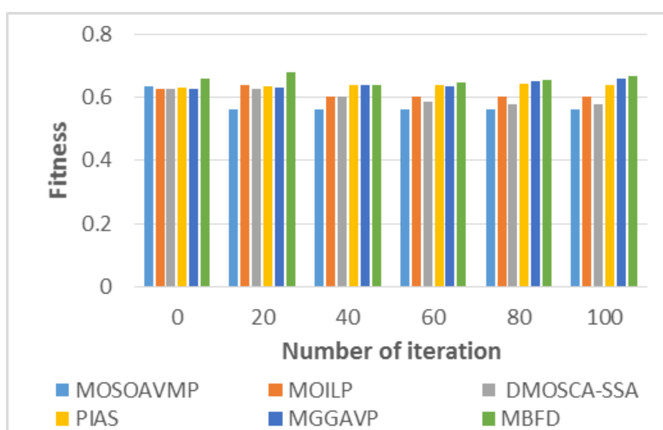

Figure 15 Convergence Time

RESEARCH ARTICLE

Figure 16 shows the comparison between the time of execution of the method that is proposed and the state methods for finding an optimal solution for VMP. To carry out this comparison, a range from 100VMs to 500VMs is used to see the evolution of execution time. Referring to Figure 15, MOSOAVMP is better at finding the optimal VMP solution at different problem sizes than other algorithms.

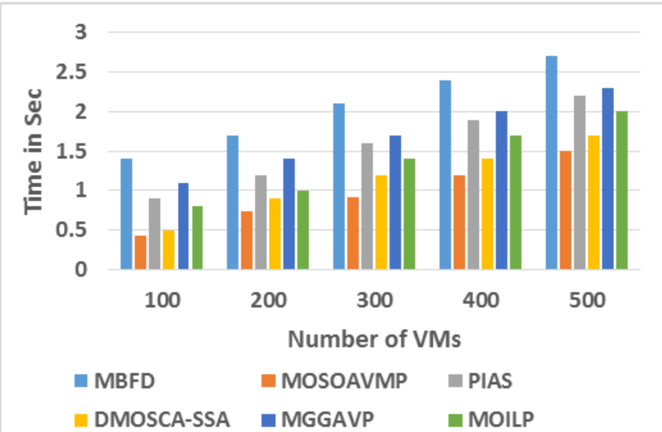

Figure 16 Average Storage Usage

5.2. Discussion and Limitation

The results show that the MOSOAVMP algorithm outperforms other leading algorithms in terms of energy consumption and resource management in cloud-based data centers. As shown in Figure 56, energy consumption increases proportionally with the number of active virtual machines (VMs). However, MOSOAVMP reduces energy consumption by 5.44% on average compared to DMOSCA-SSA, MOILP, PIAS, and MGGAVP, notably with a 6.18% improvement for 200 VMs.

In terms of VM migrations (Figure 57), the proposed algorithm demonstrates superior stability by minimizing the number of migrations required. By taking resource correlation into account, MOSOAVMP optimizes VM placement, reducing the number of active physical servers and associated costs. For example, for 2000 VMs, MOSOAVMP improves efficiency by 1.22% over DMOSCA-SSA and by 8.10% over MBFD.

Figures 59 to 62 illustrate that MOSOAVMP optimizes the use of PM resources such as CPU, memory, storage, and bandwidth, with respective improvements of 14.84%, 11.54%, 5.37%, and 6.88% compared to other algorithms. This efficient resource management is crucial to reducing waste and improving overall cloud computing performance.

When it comes to meeting SLAs, MOSOAVMP shows increased efficiency (Figure 64), reducing the fluctuations observed with other algorithms and guaranteeing a more reliable user experience. Finally, Figure 65 shows that MOSOAVMP converges faster to an optimal solution by increasing the number of iterations, demonstrating its robustness in solving complex VM placement problems.

The MOSOAVMP algorithm proves to be a superior solution for minimizing energy consumption, optimizing resources, and ensuring effective SLA management in cloud environments, systematically outperforming the other algorithms. Although the proposed method offers all these advantages, some improvements can be made by enhancing evaluation metrics such as workload handling, flexibility, adaptability, and security. These metrics, which have not been considered in this work, may impact QoS, highlighting a weakness of the proposed approach.

6. CONCLUSION

Cloud data centers consist of various power-intensive technologies, such as physical servers, switching devices, and cooling systems. However, the power consumption of these devices raises operational expenses and can lead to significant heat generation issues. In this paper, we propose an algorithm that can help optimize the VMP, intending to reduce energy consumption, enhance resource utilization, and minimize service level agreement violations. We formulate this problem as a multi-purpose problem optimization and employ a seagull-based approach for its solution. The test results show that the proposed solution achieves all the objectives set out in this work, compared with the algorithms of existing works. MOSOAVMP showed good exploitation and exploration performance. Regarding accuracy and speed of finding the best solution, MOSOAVMP has a better convergence speed.

In future research, we plan to compare this method with new metaheuristic algorithms such as [43], [44], [45], [46], [47], while considering additional metrics such as cloud security and task scheduling. Additionally, our objectives include exploring the placement of containers in the cloud using metaheuristics and deep learning. Furthermore, we will investigate the integration of game theory to enhance the optimization process.

REFERENCES

[1] M. Masdari and M. Zangakani, “Green Cloud Computing Using Proactive Virtual Machine Placement: Challenges and Issues,” J Grid Comput, 2019, doi: 10.1007/s10723-019-09489-9.
[2] J. Singh and N. K. Walia, “A Comprehensive Review of Cloud Computing Virtual Machine Consolidation,” IEEE Access, vol. 11. Institute of Electrical and Electronics Engineers Inc., pp. 106190–106209, 2023. doi: 10.1109/ACCESS.2023.3314613.
[3] A. Ndayikengurukiye, A. Ez-Zahout, A. Aboubakr, Y. Charkaoui, and O. Fouzia, “Resource Optimisation in Cloud Computing: Comparative Study of Algorithms Applied to Recommendations in a Big Data Analysis Architecture,” Journal of Automation, Mobile Robotics and Intelligent Systems, vol. 2021, no. 4, pp. 65–75, 2021, doi: 10.14313/JAMRIS/4-2021/28.
[4] S. Gharehpasha, M. Masdari, and A. Jafarian, “Power efficient virtual machine placement in cloud data centers with a discrete and chaotic

RESEARCH ARTICLE

hybrid optimization algorithm," Cluster Comput, vol. 24, no. 2, pp. 1293–1315, Jun. 2021, doi: 10.1007/s10586-020-03187-y.

[5] R. Regaieg, M. Koubàa, Z. Ales, and T. Aguili, "Multi-objective optimization for VM placement in homogeneous and heterogeneous cloud service provider data centers," Computing, vol. 103, no. 6, pp. 1255–1279, Jun. 2021, doi: 10.1007/s00607-021-00915-z.

[6] K. RahimiZadeh and A. Dehghani, "Design and evaluation of a joint profit and interference-aware VMs consolidation in IaaS cloud datacenter," Cluster Comput, vol. 24, no. 4, pp. 3249–3275, Dec. 2021, doi: 10.1007/s10586-021-03310-7.

[7] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, May 2012, doi: 10.1016/j.future.2011.04.017.

[8] Z. Xiao, J. Jiang, Y. Zhu, Z. Ming, S. Zhong, and S. Cai, "A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory," Journal of Systems and Software, vol. 101, pp. 260–272, Mar. 2015, doi: 10.1016/j.jss.2014.12.030.

[9] J. Lu, W. Zhao, H. Zhu, J. Li, Z. Cheng, and G. Xiao, "Optimal machine placement based on improved genetic algorithm in cloud computing," Journal of Supercomputing, vol. 78, no. 3, pp. 3448–3476, Feb. 2022, doi: 10.1007/s11227-021-03953-8.

[10] L. Caviglione, M. Gaggero, M. Paolucci, and R. Ronco, "Deep reinforcement learning for multi-objective placement of virtual machines in cloud datacenters," Soft comput, vol. 25, no. 19, pp. 12569–12588, Oct. 2021, doi: 10.1007/s00500-020-05462-x.

[11] F. Alharbi, Y. C. Tian, M. Tang, M. H. Ferdaus, W. Z. Zhang, and Z. G. Yu, "Simultaneous application assignment and virtual machine placement via ant colony optimization for energy-efficient enterprise data centers," Cluster Comput, vol. 24, no. 2, pp. 1255–1275, Jun. 2021, doi: 10.1007/s10586-020-03186-z.

[12] M. Ghetas, "A multi-objective Monarch Butterfly Algorithm for virtual machine placement in cloud computing," Neural Comput Appl, vol. 33, no. 17, pp. 11011–11025, Sep. 2021, doi: 10.1007/s00521-020-05559-2.

[13] S. Y. Rashida, M. Sabaei, M. M. Ebadzadeh, and A. M. Rahmani, "A memetic grouping genetic algorithm for cost efficient VM placement in multi-cloud environment," Cluster Comput, vol. 23, no. 2, pp. 797–836, Jun. 2020, doi: 10.1007/s10586-019-02956-8.

[14] H. M. R. Al-Khafaji, "Improving Quality Indicators of the Cloud-Based IoT Networks Using an Improved Form of Seagull Optimization Algorithm," Future Internet, vol. 14, no. 10, Oct. 2022, doi: 10.3390/fi14100281.

[15] S. Nabavi, L. Wen, S. S. Gill, and M. Xu, "Seagull optimization algorithm based multi-objective VM placement in edge-cloud data centers," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 28–36, 2023, doi: 10.1016/j.iotcps.2023.01.002.

[16] G. Dhiman, M. Garg, A. Nagar, V. Kumar, and M. Dehghani, "A novel algorithm for global optimization: Rat Swarm Optimizer," J Ambient Intell Humaniz Comput, vol. 12, no. 8, pp. 8457–8482, Aug. 2021, doi: 10.1007/s12652-020-02580-0.

[17] S. Mohapatra and P. Mohapatra, "American zebra optimization algorithm for global optimization problems," Sci Rep, vol. 13, no. 1, Dec. 2023, doi: 10.1038/s41598-023-31876-2.

[18] Z. Ma, G. Wu, P. N. Suganthan, A. Song, and Q. Luo, "Performance assessment and exhaustive listing of 500+ nature-inspired metaheuristic algorithms," Swarm Evol Comput, vol. 77, Mar. 2023, doi: 10.1016/j.swevo.2023.101248.

[19] R. Boopathi and E. S. Samundeeswari, "Amended Hybrid Scheduling for Cloud Computing with Real-Time Reliability Forecasting," International Journal of Computer Networks and Applications, vol. 10, no. 3, pp. 310–324, May 2023, doi: 10.22247/ijcna/2023/221887.

[20] P. Jain and S. K. Sharma, "A Load Balancing Aware Task Scheduling using Hybrid Firefly Salp Swarm Algorithm in Cloud Computing," International Journal of Computer Networks and Applications, vol. 10, no. 6, pp. 914–933, Nov. 2023, doi: 10.22247/ijcna/2023/223686.

[21] G. Dhiman and V. Kumar, "Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems," Knowl Based Syst, vol. 165, pp. 169–196, Feb. 2019, doi: 10.1016/j.knosys.2018.11.024.

[22] V. Kumar, Di. Kumar, M. Kaur, Di. Singh, S. A. Idris, and H. Alshazly, "A Novel Binary Seagull Optimizer and its Application to Feature Selection Problem," IEEE Access, vol. 9, pp. 103481–103496, 2021, doi: 10.1109/ACCESS.2021.3098642.

[23] G. Dhiman et al., "EMoSOA: a new evolutionary multi-objective seagull optimization algorithm for global optimization," International Journal of Machine Learning and Cybernetics, vol. 12, no. 2, pp. 571–596, Feb. 2021, doi: 10.1007/s13042-020-01189-1.

[24] F. M. Alamgir and M. S. Alam, "An artificial intelligence driven facial emotion recognition system using hybrid deep belief rain optimization," Multimed Tools Appl, vol. 82, no. 2, pp. 2437–2464, Jan. 2023, doi: 10.1007/s11042-022-13378-x.

[25] Q. Wang, M. Zhang, and S. Abdolhosseinzadeh, "Application of modified seagull optimization algorithm with archives in urban water distribution networks: Dealing with the consequences of sudden pollution load," Heliyon, vol. 10, no. 3, p. e24920, Feb. 2024, doi: 10.1016/j.heliyon.2024.e24920.

[26] P. Krishnadoss, V. K. Poornachary, P. Krishnamoorthy, and L. Shanmugam, "Improvised Seagull Optimization Algorithm for Scheduling Tasks in Heterogeneous Cloud Environment," Computers, Materials and Continua, vol. 74, no. 2, pp. 2461–2478, 2023, doi: 10.32604/cmc.2023.031614.

[27] X. Liu, G. Li, and P. Shao, "A Multi-Mechanism Seagull Optimization Algorithm Incorporating Generalized Opposition-Based Nonlinear Boundary Processing," Mathematics, vol. 10, no. 18, Sep. 2022, doi: 10.3390/math10183295.

[28] B. B. Al-Onazi, H. Alshamrani, F. O. Aldaajeh, A. S. A. Aziz, and M. Rizwanullah, "Modified Seagull Optimization with Deep Learning for Affect Classification in Arabic Tweets," IEEE Access, vol. 11, pp. 98958–98968, 2023, doi: 10.1109/ACCESS.2023.3310873.

[29] I. Abdullaev, N. Prodanova, K. A. Bhaskar, E. L. Lydia, S. Kadry, and J. Kim, "Task Offloading and Resource Allocation in IoT Based Mobile Edge Computing Using Deep Learning," Computers, Materials and Continua, vol. 76, no. 2, pp. 1463–1477, Aug. 2023, doi: 10.32604/cmc.2023.038417.

[30] J. Xue, X. Liu, H. Xu, and D. Zhang, "Research on the seagull optimization algorithm-based convolutional neural network rolling bearing fault diagnosis method," Engineering Research Express, vol. 5, no. 3, Sep. 2023, doi: 10.1088/2631-8695/acf09a.

[31] Q. Xia, Y. Ding, R. Zhang, H. Zhang, S. Li, and X. Li, "Optimal Performance and Application for Seagull Optimization Algorithm Using a Hybrid Strategy," Entropy, vol. 24, no. 7, Jul. 2022, doi: 10.3390/e24070973.

[32] E. S. Ghith and F. A. A. Tolba, "Tuning PID Controllers Based on Hybrid Arithmetic Optimization Algorithm and Artificial Gorilla Troop Optimization for Micro-Robotics Systems," IEEE Access, vol. 11, pp. 27138–27154, 2023, doi: 10.1109/ACCESS.2023.3258187.

[33] M. Hanif, N. Mohammad, K. Biswas, and B. Harun, "Seagull Optimization Algorithm for Solving Economic Load Dispatch Problem," in 3rd International Conference on Electrical, Computer and Communication Engineering, ECCE 2023, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ECCE57851.2023.10101516.

[34] Nair, Dhanya G., KP Sanal Kumar, and S. Anu H. Nair. "Automated Social Distance Recognition and Classification using Seagull Optimization Algorithm with Multilayer Perceptron." Dandao Xuebao/Journal of Ballistics 35.3 (2023): 11-24.

[35] A. Mohammadzadeh, D. Javaheri, and J. Artin, "Chaotic hybrid multi-objective optimization algorithm for scientific workflow scheduling in multisite clouds," Journal of the Operational Research Society, 2023, doi: 10.1080/01605682.2023.2195426.

[36] S. Kaushaley, B. Shaw, and J. Ranjan Nayak, "Optimized Machine Learning based forecasting model for Solar Power Generation by using Crow Search Algorithm and Seagull Optimization Algorithm," 2022, doi: 10.21203/rs.3.rs-1987438/v1.

RESEARCH ARTICLE

[37] A. Ndayikengurukiye, A. Ez-Zahout, and F. Omary, “An overview of the different methods for optimizing the virtual resources placement in the Cloud Computing,” J Phys Conf Ser, vol. 1743, p. 012030, Jan. 2021, doi: 10.1088/1742-6596/1743/1/012030.

[38] S. Omer, S. Azizi, M. Shojafar, and R. Tafazolli, “A priority, power and traffic-aware virtual machine placement of IoT applications in cloud data centers,” Journal of Systems Architecture, vol. 115, May 2021, doi: 10.1016/j.sysarc.2021.101996.

[39] S. Azizi, M. Shojafar, J. Abawajy, and R. Buyya, “GRVMP: A Greedy Randomized Algorithm for Virtual Machine Placement in Cloud Data Centers,” IEEE Syst J, vol. 15, no. 2, pp. 2571–2582, Jun. 2021, doi: 10.1109/JSYST.2020.3002721.

[40] S. Farzai, M. H. Shirvani, and M. Rabbani, “Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters,” Sustainable Computing: Informatics and Systems, vol. 28, Dec. 2020, doi: 10.1016/j.suscom.2020.100374.

[41] T. B. Hewage, S. Ilager, M. A. Rodriguez, and R. Buyya, “CloudSim express: A novel framework for rapid low code simulation of cloud computing environments,” Softw Pract Exp, 2023, doi: 10.1002/spe.3290.

[42] I. Behera and S. Sobhanayak, “Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach,” J Parallel Distrib Comput, vol. 183, Jan. 2024, doi: 10.1016/j.jpdc.2023.104766.

[43] Y. Yuan et al., “Coronavirus Mask Protection Algorithm: A New Bio-inspired Optimization Algorithm and Its Applications,” J Bionic Eng, vol. 20, no. 4, pp. 1747–1765, Jul. 2023, doi: 10.1007/s42235-023-00359-5.

[44] F. A. Zeidabadi, M. Dehghani, P. Trojovský, Š. Hubálovský, V. Leiva, and G. Dhiman, “Archery Algorithm: A Novel Stochastic Optimization Algorithm for Solving Optimization Problems,” Computers, Materials and Continua, vol. 72, no. 1, pp. 399–416, 2022, doi: 10.32604/cmc.2022.024736.

[45] W. Zilong and S. Peng, “A Multi-Strategy Dung Beetle Optimization Algorithm for Optimizing Constrained Engineering Problems,” IEEE Access, vol. 11, pp. 98805–98817, 2023, doi: 10.1109/ACCESS.2023.3313930.

[46] Y. Du, H. Yuan, K. Jia, and F. Li, “Research on Threshold Segmentation Method of Two-Dimensional Otsu Image Based on Improved Sparrow Search Algorithm,” IEEE Access, vol. 11, pp. 70459–70469, 2023, doi: 10.1109/ACCESS.2023.3293191.

[47] Aristide Ndayikengurukiye, Abderrahmane Ez-Zahout, Fouzia Omary, "Optimizing Virtual Machines Placement in a Heterogeneous Cloud Data Center System", International Journal of Computer Networks and Applications (IJCNA), 11(1), PP: 1-12, 2024, DOI: 10.22247/ijcna/2024/224431.

Authors

Aristide Ndayikengurukiye graduated with an Msc in Computer Engineering from the University of Burundi in 2017. PhD Student in Intelligent Processing Systems & Security Team (IPSS), Computer Science Department Faculty of Science of Rabat, Mohammed V University, Morocco. His research interests include Cloud computing optimization, artificial intelligence, and Big Data.

Rim Doukha is currently a Ph.D. student in computer science at the Department of Computer Science, Faculty of Science, Mohammed V University. Her research focuses on the fields of cloud computing and deep learning. She holds a Master's degree in Internet of Things and Mobile Services from ENSIAS. Notably, she has completed several internships in Morocco and France, worked in Belgium as a Cloud Computing research engineer as part of a European project, and is currently a project officer in cloud computing and technical standardization in Luxembourg.

Eric Niyukuri is currently a software developer at Orion Systems & Design. He has been working in the FinTech industry, especially on electronic payment systems such as EFT, RTGS and Cheque Truncation systems, Mobile Banking, and E-Tax systems. Before joining the FinTech industry he developed software for insurance. He holds a Master of Science degree in Computer Engineering from the University of Burundi.

Eric Muheto As a seasoned Unix SME at Morgan Stanley, I excel in DevOps and middleware, holding a computer science degree and Lib@Web certification. Proficient in Unix and cloud management with tools like Aquilon and Ansible, I specialize in automation using Python scripts. Notable projects include enhancing safety at Canadian National Railways and containerizing legacy apps with Docker. Key contributions at the National Bank of Canada.

Abderrahmane Ez-Zahout is a Professeur Habilité of computer science at the Department of Computer Science/faculty of Sciences at Mohammed V University. His research is in the fields of computer sciences, digital systems, big data, and computer vision. Recently, he has worked on intelligent systems. He has served as an invited reviewer for many journals. Besides, he is also involved in NGOs, and student associations, and managing non-profit foundations.

Fouzia Omary is the leader of IPSS team research and is currently a full professor of computer science at the Department of Computer Science/ENS, Mohammed V University. She is a full-time researcher at Mohammed V University and she is the chair of the international conference on cybersecurity and blockchain and also the leader of the national network of blockchain and cryptocurrency.

How to cite this article:

Aristide Ndayikengurukiye, Rim Doukha, Eric Niyukuri, Eric Muheto, Abderrahmane Ez-zahout, Fouzia Omary, “SOAVMP: Multi-Objective Virtual Machine Placement in Cloud Computing Based on the Seagull Optimization Algorithm”, International Journal of Computer Networks and Applications (IJCNA), 11(3), PP: 375-389, 2024, DOI: 10.22247/ijcna/2024/24.