Black Widow Optimization Algorithm for Virtual Machines Migration in the Cloud Environments

Chuang Zhou[1*]
Anhui Vocational College of Defense Technology
Lu'an, Anhui, 237011, China

***Abstract*—Cloud data centers use virtualization technology to manage computing resources. Using a group of connected Virtual Machines (VMs), corresponding users can compute data efficiently and effectively. It improves the utilization of resources, thereby reducing hardware requirements. Repossession of affected services requires VM-based infrastructure overhaul schemes. Clarifications concerning dedicated routing are also desirable to improve the reliability of Domain Controller (DC) services. The migration of a VM experiencing a node failure challenges maintaining reliability. The selection of VMs is influential in limiting the number of VM migrations. Choosing one or more potential VMs for migration reduces the servers' workload. This paper presents an energy-aware VM migration method for cloud computing based on the Black Widow Optimization (BWO) algorithm. The proposed algorithm was implemented and measured using JAVA. Afterward, we compared our results against existing methodologies regarding resource availability, energy consumption, load, and migration cost.**

Keywords*—Cloud computing; migration; energy consumption; optimization; black widow algorithm

I. Introduction

Cloud computing is rapidly consolidating itself as a new computing technology [1]. It aims to provide services and meet user demands for high reliability, scalability, and availability [2]. Cloud computing has gained popularity over the past few years due to its efficient utilization of resources and convenience in accessing services [3]. These competitive advantages can be attributed to virtual technology and distributed networking in the cloud [4]. Storage, processing, memory, bandwidth, network, and virtual machines are some of the resources provided by cloud computing [5]. The provided virtualized services can be categorized as Platform as a Service (PaaS), Infrastructure as a Service (IaaS), and Software as a Service (SaaS) [6]. Virtualization facilitates access to resources while hiding their physical characteristics [7]. This technique allows several different environments to be separated or shared so that they can interact without being aware of one another [8]. As a matter of fact, virtualization is the key technology behind cloud computing. Data centers typically consist of hundreds of heterogeneous servers that consume significant energy. Increasing emissions and climate change have prompted governments, organizations, and IT enterprises to try and manage data centers more sustainably [9]. Consequently, data centers in the cloud require considerable energy as cloud services grow rapidly [10]. Cloud service providers face a serious challenge in reducing energy consumption. Two factors cause this situation. The first is that data centers consume a lot of energy, resulting in carbon dioxide emissions that are incompatible with the environment. The second reason is the low power efficiency [11].

The concept of migrating Virtual Machines (VMs) dynamically and transparently from one host to another is called VM migration [12]. In addition, VM migration provides an opportunity to identify hotspots in data centers. In the realm of virtual machine migration within cloud environments, the convergence of IoT, Artificial Intelligence (AI), and Machine Learning (ML) introduces transformative capabilities. IoT sensors enable real-time data collection, facilitating dynamic workload monitoring and resource utilization assessment [13-15]. AI algorithms harness this influx of data to make intelligent decisions on VM migrations, optimizing resource allocation and energy efficiency. ML algorithms, fueled by historical and real-time data, identify patterns and predict workload trends, enabling proactive migration strategies to prevent bottlenecks and ensure optimal performance [16, 17]. The synergy of IoT, AI, and ML empowers cloud systems to autonomously adapt to fluctuating workloads, enhance resource provisioning, and optimize VM migrations, thereby elevating the efficiency and resilience of cloud environments.

Workload regulation on individual nodes via VM migration to optimize energy consumption is a Non-deterministic Polynomial (NP)-hard problem, commonly solved with heuristic approaches. In these cases, metaheuristic algorithms can be used efficiently when local heuristics are insufficient for finding optimum solutions [18]. Metaheuristics are repetitive procedures used as guides and amendments to other heuristics. Based on the Black Widow Optimization (BWO) algorithm, the current study proposes a new VM migration method that reduces energy consumption, hosts, and migrations. The BWO has a short computation time for globally optimized results, a high convergence speed as the quality parameters can be regulated, and a flexible optimization pattern [19]. This algorithm draws inspiration from the distinctive mating conduct observed in black widow spiders. Owing to its unique operators, it can be deemed a fusion of evolutionary and swarm methodologies. Notably, the BWO algorithm introduces a distinctive phase called "cannibalism." This phase holds a significant advantage as it eliminates species exhibiting inadequate fitness from the population, thus fostering an accelerated convergence rate. This paper contributes to the following:

- Multi-objective VM migration model designed for choosing the most suitable VM.
- The BWO algorithm is proposed to select the most efficient VMs in the cloud.

The rest of the paper is organized as follows. The literature related to this topic is briefly reviewed in Section II. Section III describes the proposed algorithm. The experiments and results are summarized in Section IV. Section V outlines the conclusions derived from the analysis.

II. Related work

Kansal and Chana [20] have proposed a method for migrating VMs based on Firefly Optimization (FFO). The FFO algorithm optimizes effective energy simultaneously at the memory and processor levels. Furthermore, it reduces the number of PMs and VMs, which avoids further energy wastage. VMs with the highest load are transmitted to lower-load nodes to maintain efficiency and performance. Wang, et al. [21] introduced enough green energy-efficient data centers. Diverse renewable energy supplies have been considered in the efficient management of VM migration. The proposed method can flexibly manage green energy and cooling power consumption. The authors assessed the effect of temperature on the energy consumption of cooling and IT devices.

Xu and Abnoosian [12] presented a hybrid optimization algorithm based on genetic and particle swarm optimization algorithms for improving VM energy consumption and execution time during VM migration. In the hybrid algorithm, GA is utilized to overcome the limitations of the PSO algorithm, which suffers from slow convergence and limited global optimization. According to the results, the proposed method has improved energy consumption by an average of 23.19% compared to the other three methods. Results also revealed a 29.01% improvement in execution time over the other three methods. Zhou, et al. [22] introduce an energy-efficient algorithm for VM migrations. This algorithm optimizes host location, VM selection, and trigger time when memory and CPU factors are considered. It migrates some VMs from lightly loaded to heavily loaded hosts using virtualization technology. Energy is conserved by switching idle hosts to the low-power mode or shutting them down. This algorithm reduces SLA violations by 13% and saves 7% of energy over the Double Threshold (DT) algorithm.

Fu, et al. [23] presented a layered VM migration algorithm. Cloud data centers are divided into several regions based on bandwidth utilization rates. VM migrations balance network load between regions, resulting in load balancing of cloud resources. Experiments indicate that the proposed algorithm is shown to be able to balance network resource load in cloud computing effectively. Chien, et al. [24] have proposed an efficient VM migration algorithm based on minimizing migrations in cloud computing to improve efficiency, meet user requirements, and prevent service level agreements (SLA) violations. The proposed algorithm was more effective than existing algorithms based on experimental results. A threshold algorithm was proposed by Kaur and Sachdeva [25] to allocate tasks to the most capable machine and host and to maintain checkpoints on VMs. Overloaded VMs need to migrate tasks to another VM. This study proposes a weight-based technique for migrating cloudlets between VMs.

Cao and Hou [26] have introduced a bi-level VM placement algorithm. The initial tier incorporates a queuing model devised to manage a multitude of VM requests. This model facilitates the seamless implementation and validation of diverse models, including cloud simulations. Additionally, it furnishes an alternative mechanism for task allocation to servers. Subsequently, a multi-objective VM placement algorithm is introduced, rooted in the Krill Herd (KH) algorithm. Fundamentally, this algorithm strives to strike an equilibrium between energy consumption and the efficient utilization of resources.

Khan and Santhosh [27] have proposed a hybrid optimization algorithm for managing VM migration within a cloud environment. This novel approach amalgamates Particle Swarm Optimization (PSO) and Cuckoo Search (CS) algorithms to yield the proposed hybrid optimization framework. The focal point of this research endeavor is mitigating energy consumption, computation time, and migration expenses. Additionally, a secondary objective pertains to the maximization of resource utilization. To substantiate the research objectives, the efficacy of the hybrid optimization model is rigorously assessed through simulation analysis. This assessment encompasses a comparative study against conventional algorithms. The evaluation parameters encompass computation time, resource availability, migration cost, and energy consumption.

Kumar and Sivakumar [28] have introduced a novel approach for VM migration, hinging on the CS algorithm. The selection of an appropriate provider is undertaken through a comprehensive consideration of multiple constraints, including factors such as delay, bandwidth, cost, and load. Subsequent to this, effective search criteria are calculated, facilitating the identification of the optimal service contingent upon fitness constraints. These search criteria are framed as optimization problems, and their resolution is undertaken using the CS algorithm. The proposed CS algorithm is meticulously designed by integrating the Cuckoo Search Optimization (CSO) technique with the Salp Swarm Algorithm (SSA). This amalgamation ensures the evaluation of the fitness function for optimal VM migration, incorporating diverse parameters encompassing delay, cost, bandwidth, and load. Consequently, the cloud manager can adeptly execute VM migration in the cloud environment, leveraging the proposed CS-based VM migration approach. The performance evaluation of the CS-based VM migration technique focuses on delay, cost, and load. The results demonstrate that the proposed CS-based VM migration methodology achieves commendable outcomes, manifesting in a minimal delay of 0.14, cost of 0.05, and load of 0.18.

These studies address various facets of the problem, each contributing unique insights and methodologies. Notably, challenges related to optimizing energy efficiency, load balancing, resource utilization, and cost reduction are prevalent across these studies. Kansal and Chana [20] focus on optimizing energy efficiency and reducing the number of PMs and VMs. Wang, et al. [21] explore the integration of diverse renewable energy supplies and cooling power consumption management. Xu and Abnoosian [12] tackle slow convergence and limited global optimization issues in hybrid optimization. Zhou, et al. [17 optimize host location, VM selection, and trigger time for energy-efficient migrations. Fu, et al. [23] propose a layered VM migration algorithm for network load balancing. Chien, et al. [24] emphasize minimizing migrations to prevent SLA violations. Kaur and Sachdeva [25] propose a threshold algorithm for efficient task allocation. Cao and Hou [26] address energy consumption and resource utilization equilibrium. Khan and Santhosh [27] and Kumar and Sivakumar [28] introduce hybrid optimization techniques to mitigate energy consumption and migration costs, and enhance resource utilization while considering multiple constraints. These studies underscore the multifaceted nature of VM migration optimization and lay the groundwork for further advancements in this evolving field.

III. Proposed Method

In this section, we discuss virtual machine migration using the BWO. The cloud migration model is illustrated in Fig. 1. Cloud models involve multiple PMs for handling user requests, and PMs gather the VMs to perform tasks on demand. The VMs are created dynamically to alleviate the bottlenecks in cloud computing, and virtualization enhances speed. Cloud services are delivered as tasks to users and are assigned to VMs in a round-robin fashion. In this case, PM controls the set of VMs, and a load balancer monitors loads of PMs. VM migration occurs when a load of a PM exceeds a threshold level. The analytical approaches employed in the development of the VM migration model are as follows.

- A cloud with m PMs and n VMs is created initially.
- This stage sets the migration cost of PMs to the highest value, so it equals 1.
- Round-robin assignment is used to assign incoming tasks to VM at the time interval.
- When the VM's load value exceeds the threshold, migrate the VM in an optimal manner using the proposed BWO.
- This step determines qualitative criteria, such as energy consumption, resource availability, and migration costs.
- The algorithm repeats steps (3) to (5) for every iteration, and then it ends.

A. Initialization

Our cloud system consists of n VMs and m PMs. According to Eq. 1, C, PM_1, and PM_m stand for cloud, the first PM, and m^{th} PM, respectively.

$$C = \{PM_1, PM_2, \dots, PM_m\} \tag{1}$$

Eq. 2 illustrates VMs, with VM_1 being the first one and VM_n being the last one.

$$PM_m = \{VM_1, VM_2, \dots, VM_n\} \tag{2}$$

The cloud is populated by t users involving k tasks. Eq. 3 can be used to indicate each user.

$$U_t = \{T_1, T_2, \dots, T_k\} \tag{3}$$

In addition, a round-robin process assigns the users' tasks to VMs. In the cloud model, VMs are determined by several parameters, such as memory, bandwidth, CPU, and processing power. VMs in cloud computing environments have the following characteristics, as defined by Eq. 4.

$$V_m^n = \{I_m^n, D_m^n, B_m^n, C_m^n, J_m^n\} \tag{4}$$

I_n^m denotes the total number of utilized MIPS, D_n^m refers to memory, B_n^m signifies the bandwidth, C_n^m represents the number of CPUs, and J_n^m stands for the total number of processing entities. A scale of 1 to 10 is given for the above parameters.

B. Load computation

The load is calculated based on the resources needed by VMs to process tasks supplied by the user. A cloud load is evaluated by considering the processing power, CPU, memory, MIPS, and bandwidth using Eq. 5, in which t represents time, and RU stands for resource utilization. Eq. 6 calculates the resource used by m^{th} VM present. The variables in Eq. 6 are defined in Table I.

$$Load(L) = \frac{R_U}{t} \tag{5}$$

$$R_U = \frac{1}{F}\sum_{i=1}^{M}\left(\frac{I^F}{\max(I^F)} + \frac{D^F}{\max(D^F)} + \frac{B^F}{\max(B^F)} + \frac{C^F}{\max(C^F)} + \frac{J^F}{\max(J^F)}\right) \tag{6}$$

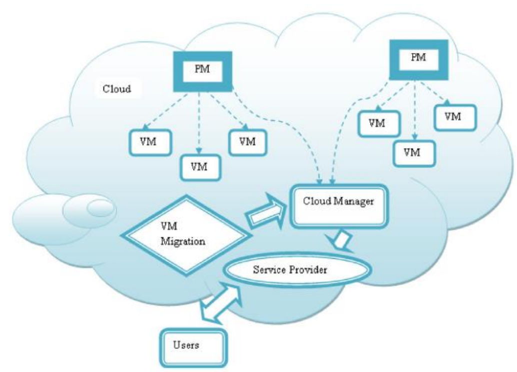

Fig. 1. VM migration model.

TABLE I. VARIABLES IN EQ. 6.

Variable	Definition
F	Normalization factor
m	Number of VMs in each PM
I^F	MIPS
D^F	Memory
B^F	Bandwidth
C^F	CPU
J^F	Processing entity

C. Resource Availability

It is responsible for ensuring the efficient use of resources. Load balancing must be optimized for well-organized performance. Eq. 7 calculates resource availability.

$$R_A = 1 - R_U \tag{7}$$

D. Migration Cost

A VM migration cost is determined by the number of movements. Migration costs for the entire cloud environment can be calculated by Eq. 8, in which *c* stands for constant, *M* refers to the total number of VMs, *G* denotes the number of migrations, and *N* is the total number of PMs.

$$M_c = \frac{1}{N}\sum_{j=1}^{n}\left(\frac{G}{c \times M}\right) \tag{8}$$

E. Energy Model

Each VM consumes energy to migrate and process data. In this way, cloud setup energy is primarily dependent on the power consumed by the resources within the VM, calculated by Eq. 9, in which *T* represents the total duration, and *K* denotes the power consumed by the VM calculated by Eq. 10. The maximum power consumed is denoted by K_{max}, and the resource utilization is represented by R_U^{cloud} calculated by Eq. 11.

$$E = \frac{1}{T}\sum_{t=1}^{T} K \tag{9}$$

$$K = p \times K_{max} + (1-p) \times K_{max} \times R_U^{cloud}; \quad 1 < p < t < T \tag{10}$$

$$R_U^{cloud} = \frac{1}{N}\sum_{r=1}^{N} R_U \tag{11}$$

F. Solution encoding

The proposed algorithm finds suitable VMs for migration. Consider a scenario in which PM 1 involves 2 VMs, and PM 2 involves 3 VMs. A round-robin assignment process assigns the incoming tasks to VMs. All processes are handled in a circular order, according to assigned time slices, with no priority given to any process. VMs are migrated to under-loaded PMs if they exceed a threshold value, and an optimization algorithm is used to assign the optimal VM. A solution encoding for identifying optimal VMs for migration is shown in Fig. 2.

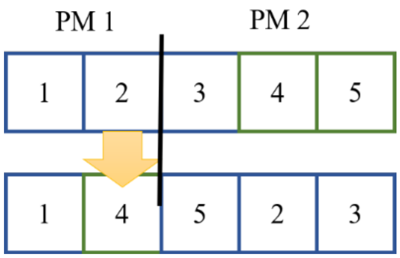

Fig. 2. Encoding solutions.

G. Fitness function

An optimal solution is determined by computing the fitness function. Energy, migration cost, load, and resource utilization are the parameters that are used in formulating the fitness function. VMs are selected based on the fitness function, calculated by Eq. 12.

$$Fitness(f) = \delta(1-E) + \gamma(1-M_c) + \beta(1-L) + aR_A \tag{12}$$

In Eq. 12, *E* stands for energy, M_C represents migration costs, *L* refers to the load, R_A denotes resource utilization, δ, γ, β, and α are the weights ranging from 0 to 1.

H. BWO algorithm for VM migration

This section applies the BWO algorithm to the multi-objective VM migration method. VMs are migrated to specific VMs based on cost, energy, and load. The performance of the VM migration algorithm is determined by the reduction in energy consumption, cost, and load. As with traditional methods, BWO starts by initializing a population of spiders representing possible solutions. New generations of spiders are produced in pairs. As part of this optimization, female spiders eat black male spiders during or after mating. Female black widow spiders release their stored sperm into egg sacs after storing them in their sperm theca. The spiderling emerges from its egg sac after 11 days. A spiderling remains in its mother's web for several days to a week, during which sibling cannibalism occurs. When the wind blows, the spiderling leaves the web.

- Initializing the population

Initially, the population is defined by the number of spiders, each representing a possible solution. An individual black widow represents a solution to an optimization problem in d-dimensional space. Eq. 13 defines the array.

$$Black\ widow = [x_1, x_2, \dots, x_d] \tag{13}$$

The array contains floating point variables. There are NP black widow spiders in a black widow population, expressed as an *NP*d* candidate matrix. Eq. 14 assigns the initial population at random.

$$Black\ widow = xl + rand(l, d) \times (xu - xl) \tag{14}$$

- Procreate

As each pair is independent of the other, it starts mating to reproduce a new generation, just as each couple naturally mates in its web, independently of the rest of the web. Even though hundreds of eggs are laid every time a spider mates, the number of muscular spider babies remains the same. The BWO produces offspring by reproducing an array called alpha using arbitrary numbers, in which u1 and u2 represent parents, whereas v1 and v2 represent children.

$$\begin{cases} v_1 = a \times u_1 + (1-a) \times u_2 \\ v_2 = a \times u_2 + (1-a) \times u_1 \end{cases} \tag{15}$$

- Cannibalism

Three categories of cannibalism are included in this algorithm. One of the earliest forms is cannibalism, where a female black widow eats a black male widow after or during mating. Fitness values are used to identify female and male black widows. Sibling cannibalism is another type of cannibalism in which stronger siblings eat weaker siblings. The algorithm sets the cannibalism rating (CR) based on survivor numbers. The third type of cannibalism occurs when the baby spider eats the mother spider. Weak or strong spiderlings are evaluated based on their fitness value.

- Mutation

Based on the simulated binary crossover (SBC), the BWO algorithm produces new chromes or individuals at constant crossover and mutation rates. A mutation rate is altered using an adaptive scheme, followed by a projection of the enhanced mutation rate. It combines three crossover operators, single-point crossover (SPC), uniform crossover (UC), and SBC, to generate new individuals. As a result, the proposed algorithm is evaluated on three variables. BWO algorithm fails to determine an optimal solution to the optimization problem based on the permanent mutation rate. With the adaptive strategy employed in this paper, the mutation rate can be altered in order to solve this problem. We present a linear function that alters the mutation rate for ease of use. It is, therefore, necessary to update the mutation rate using Eq. 16, in which *ps* is calculated by Eq. 17.

$$m_r = \frac{p_s}{L} \tag{16}$$

$$p_s = P + (t-1) \times \frac{1-P}{t_M - 1} \tag{17}$$

In Eq. 17, t_M and t stand for the maximum and current generation, respectively, P refers to the fixed real number given by Eq. 18.

$$P_0 = \frac{L}{50} \tag{18}$$

IV. Experimental Results

In this section, we present results obtained from the proposed algorithm and compare them with previous methods. The developed algorithm was tested in JAVA on a PC with Windows 8 and 8GB of RAM. Simulation is conducted in a cloud-based environment containing 10 PMs and 50 VMs, with 25 incoming tasks. The main evaluation indicators are resource availability, energy, migration cost, and load. VM migration techniques on cloud computing platforms commonly use these performance metrics. We compared WOA [29], Firefly [20], and ABC-BA [30] with our proposed algorithm for analysis.

An analysis of 25 incoming tasks is conducted by comparing the developed algorithm with respect to migration cost, resource availability, energy, and load. In Fig. 3, resource availability is compared over a variety of iterations. For the 50th iteration, WOA, Firefly, and ABC-BA have resource availability values of 0.96, 0.952, and 0.938, respectively, which are lower than our algorithm. Fig. 4 illustrates the energy cost analysis. In the 60th iteration, existing techniques, such as WOA, Firefly, and ABC-BA, possess energy costs of 0.498, 0.494, and 0.494, respectively, which are higher than our algorithm. Fig. 5 shows a comparative analysis based on migration costs. At 60 iterations, WOA, Firefly, ABC-BA, and our algorithm achieved migration cost values of 0.195, 0.132, 0.0605, and 0.059, respectively. Fig. 6 shows the results of the load analysis based on different iterations. At 50 iterations, WOA, Firefly, ABC-BA, and our algorithm have corresponding load values of 0.0098, 0.0038, 0.0029, and 0.0019.

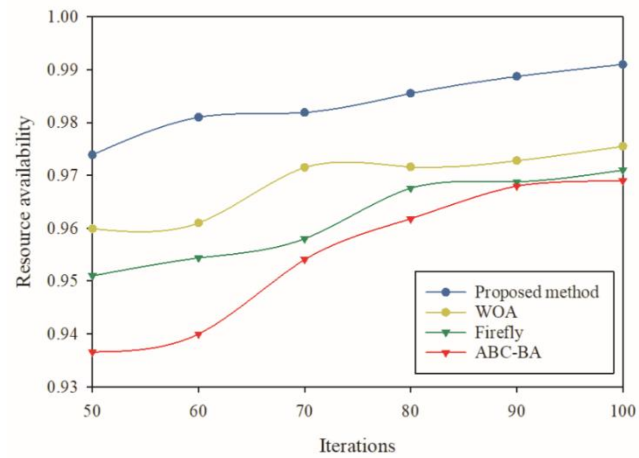

Fig. 3. Resource availability comparison.

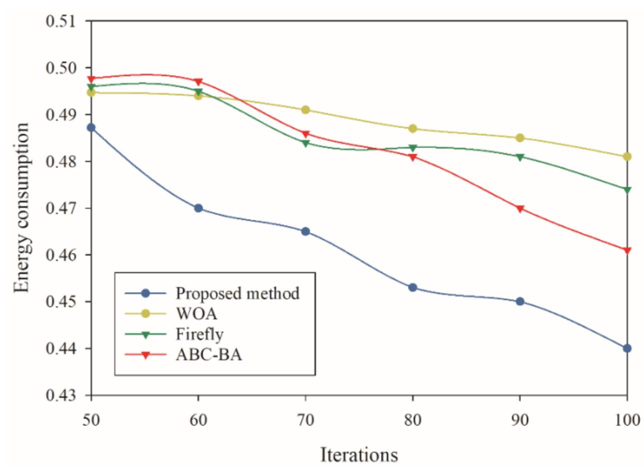

Fig. 4. Energy consumption comparison.

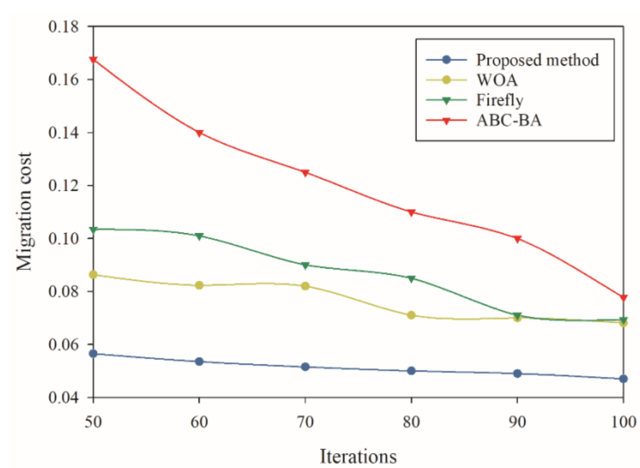

Fig. 5. Migration cost comparison.

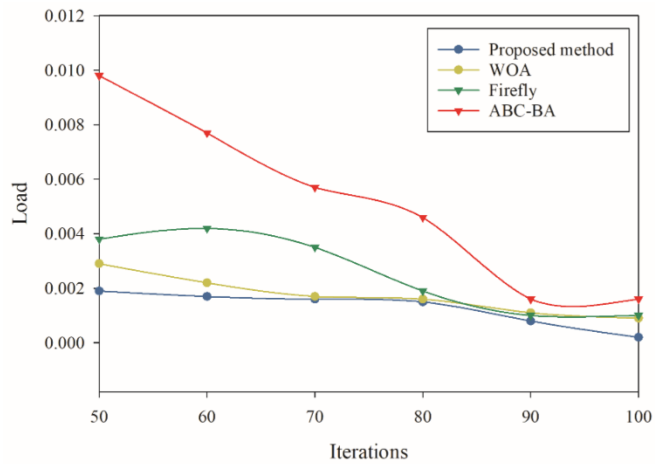

Fig. 6. Load comparison.

The datasets used in our tests were chosen with attention to replicating different characteristics of real-world cloud computing infrastructures. We simulate the complicated dynamics and complexity of existing cloud infrastructures by using datasets with 10 PMs and 50 VMs accommodating 25 incoming tasks. These datasets contain a variety of resource capabilities, imitating real-world cloud setups and emerging as a relevant depiction of dynamic workload management concerns. The intrinsic heterogeneity of these datasets, which includes the interaction of PMs, VMs, and tasks, affects the performance of VM migration algorithms. The datasets' intricacies directly influence resource use, energy consumption, migration cost, and load distribution, all of which are critical performance measures in our study. Because cloud services are resource-intensive, efficient and adaptable migration solutions are required, which our suggested algorithm tries to meet. As we examine the differences in comparing outcomes, it becomes clear that the success of our algorithm stems from its capacity to negotiate the complexities of various datasets intelligently. The algorithm's flexibility in changing situations, such as dynamic workloads and fluctuating resource availability, presents it as a reliable option for optimizing VM migrations.

V. Conclusion

Due to the inherent benefits of cloud computing, the number of cloud users and their workloads is increasing daily. Besides, service providers are challenged by maintaining QoS under heavy workloads. Cloud computing can offer better computing services by utilizing VM migration techniques, reducing delays, and optimizing energy usage. This paper successfully developed the BWO algorithm to migrate VMs to the cloud. This algorithm reduces unnecessary migrations by identifying the optimal solution for migrating VMs to the host. Based on simulation results, the proposed algorithm is more efficient regarding migration cost, resource availability, load, and energy consumption than previous methods.

Acknowledgment

Anhui Provincial Education Department Scientific Research Project (KJ2021A1491).

References

[1] B. Pourghebleh, A. A. Anvigh, A. R. Ramtin, and B. Mohammadi, "The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments," Cluster Computing, pp. 1-24, 2021.

[2] S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, "Load balancing in cloud computing environment using the Grey wolf optimization algorithm based on the reliability: performance evaluation," The Journal of Supercomputing, vol. 78, no. 1, pp. 18-42, 2022.

[3] A. Najafizadeh, A. Salajegheh, A. M. Rahmani, and A. Sahafi, "Multi-objective Task Scheduling in cloud-fog computing using goal programming approach," Cluster Computing, vol. 25, no. 1, pp. 141-165, 2022.

[4] H. Jin, S. Lv, Z. Yang, and Y. Liu, "Eagle strategy using uniform mutation and modified whale optimization algorithm for QoS-aware cloud service composition," Applied Soft Computing, vol. 114, p. 108053, 2022.

[5] M. R. Dorsala, V. Sastry, and S. Chapram, "Blockchain-based solutions for cloud computing: A survey," Journal of Network and Computer Applications, vol. 196, p. 103246, 2021.

[6] J. Yang, B. Jiang, Z. Lv, and K.-K. R. Choo, "A task scheduling algorithm considering game theory designed for energy management in cloud computing," Future Generation computer systems, vol. 105, pp. 985-992, 2020.

[7] Y. Wang, J. Wen, Q. Wu, L. Guo, and B. Tao, "A dynamic cloud service selection model based on trust and SLA in cloud computing," International Journal of Grid and Utility Computing, vol. 10, no. 4, pp. 334-343, 2019.

[8] X. Wei, "Task scheduling optimization strategy using improved ant colony optimization algorithm in cloud computing," Journal of Ambient Intelligence and Humanized Computing, pp. 1-12, 2020.

[9] Y. Karaca, M. Moonis, Y.-D. Zhang, and C. Gezgez, "Mobile cloud computing based stroke healthcare system," International Journal of Information Management, vol. 45, pp. 250-261, 2019.

[10] A. Rezaeipanah, M. Mojarad, and A. Fakhari, "Providing a new approach to increase fault tolerance in cloud computing using fuzzy logic," International Journal of Computers and Applications, vol. 44, no. 2, pp. 139-147, 2022.

[11] G. J. Ibrahim, T. A. Rashid, and M. O. Akinsolu, "An energy efficient service composition mechanism using a hybrid meta-heuristic algorithm in a mobile cloud environment," Journal of parallel and distributed computing, vol. 143, pp. 77-87, 2020.

[12] Y. Xu and K. Abnoosian, "A new metaheuristic - based method for solving the virtual machines migration problem in the green cloud computing," Concurrency and Computation: Practice and Experience, vol. 34, no. 3, p. e6579, 2022.

[13] R. Singh et al., "Analysis of Network Slicing for Management of 5G Networks Using Machine Learning Techniques," Wireless Communications and Mobile Computing, vol. 2022, 2022.

[14] B. Pourghebleh and N. J. Navimipour, "Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research," Journal of Network and Computer Applications, vol. 97, pp. 23-34, 2017.

[15] P. He, N. Almasifar, A. Mehbodniya, D. Javaheri, and J. L. Webber, "Towards green smart cities using Internet of Things and optimization algorithms: A systematic and bibliometric review," Sustainable Computing: Informatics and Systems, vol. 36, p. 100822, 2022.

[16] J. Webber, A. Mehbodniya, Y. Hou, K. Yano, and T. Kumagai, "Study on idle slot availability prediction for WLAN using a probabilistic neural network," in 2017 23rd Asia-Pacific Conference on Communications (APCC), 2017: IEEE, pp. 1-6.

[17] M. Bagheri et al., "Data conditioning and forecasting methodology using machine learning on production data for a well pad," in Offshore Technology Conference, 2020: OTC, p. D031S037R002.

[18] R. W. Ahmad, A. Gani, S. H. A. Hamid, M. Shiraz, A. Yousafzai, and F. Xia, "A survey on virtual machine migration and server consolidation frameworks for cloud data centers," Journal of network and computer applications, vol. 52, pp. 11-25, 2015.

[19] V. Hayyolalam and A. A. P. Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems," Engineering Applications of Artificial Intelligence, vol. 87, p. 103249, 2020.

[20] N. J. Kansal and I. Chana, "Energy-aware virtual machine migration for cloud computing-a firefly optimization approach," Journal of Grid Computing, vol. 14, no. 2, pp. 327-345, 2016.

[21] X. Wang, Z. Du, Y. Chen, and M. Yang, "A green-aware virtual machine migration strategy for sustainable datacenter powered by renewable energy," Simulation Modelling Practice and Theory, vol. 58, pp. 3-14, 2015.

[22] Z. Zhou, J. Yu, F. Li, and F. Yang, "Virtual machine migration algorithm for energy efficiency optimization in cloud computing," Concurrency and Computation: Practice and Experience, vol. 30, no. 24, p. e4942, 2018.

[23] X. Fu, J. Chen, S. Deng, J. Wang, and L. Zhang, "Layered virtual machine migration algorithm for network resource balancing in cloud computing," Frontiers of Computer Science, vol. 12, no. 1, pp. 75-85, 2018.

[24] N. K. Chien, V. S. G. Dong, N. H. Son, and H. D. Loc, "An efficient virtual machine migration algorithm based on minimization of migration in cloud computing," in International Conference on Nature of Computation and Communication, 2016: Springer, pp. 62-71.

[25] G. Kaur and R. Sachdeva, "Virtual machine migration approach in cloud computing using genetic algorithm," in Advances in Information Communication Technology and Computing: Springer, 2021, pp. 195-204.

[26] H. Cao and Z. Hou, "Krill Herd Algorithm for Live Virtual Machines Migration in Cloud Environments," International Journal of Advanced Computer Science and Applications, vol. 14, no. 5, 2023.

[27] M. S. A. Khan and R. Santhosh, "Hybrid optimization algorithm for vm migration in cloud computing," Computers and Electrical Engineering, vol. 102, p. 108152, 2022.

[28] A. Kumar and P. Sivakumar, "Cat-squirrel optimization algorithm for VM migration in a cloud computing platform," International Journal on Semantic Web and Information Systems (IJSWIS), vol. 18, no. 1, pp. 1-23, 2022.

[29] S. Mirjalili and A. Lewis, "The whale optimization algorithm," Advances in engineering software, vol. 95, pp. 51-67, 2016.

[30] K. Karthikeyan et al., "Energy consumption analysis of Virtual Machine migration in cloud using hybrid swarm optimization (ABC–BA)," The Journal of Supercomputing, vol. 76, no. 5, pp. 3374-3390, 2020.