Transactions on Machine Intelligence 3(4) (2022) 199-210

ISSN Online: 2821-1693

Transactions on Machine Intelligence

Journal Homepage: https://tmachineintelligence.ir/index.php/TMI

A Novel Approach to Reducing Energy Consumption, Economic Savings, Service Quality Enhancement, and Resource Utilization in Cloud Data Centers

M. H. Mahmoudian[1,*], H. Taheri[2], Ali Beik-Mohammadi[1]

[1] Student of Digital Electronics Department, Faculty of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

[2] Associate Professor, Faculty of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

ARTICLE INFO

Article History:
Received 6 July 2020
Received in revised form 11 September 2020
Accepted 2 December 2020
Available online 3 December 2020

Keywords:
Virtual Machine Deployment, Optimization, Virtualization, Energy Consumption, Service Quality, Data Centers, Virtual Resources, Cloud Computing.

ABSTRACT

Cloud data centers often provide the infrastructure for millions of virtual machines in dynamic environments. Virtual machine deployment is a process where it is determined which virtual machines should be executed on which physical machines within the virtualized infrastructure. Given the randomness of customer requests, the virtual machine deployment issue must be formulated as a dynamic optimization problem. On the other hand, providers must be able to respond to virtual resource requests in complex dynamic cloud computing environments, considering service elasticity and overbooking physical resources. In this work, five experiments were designed and conducted for the two-stage optimization of such issues. In these experiments, after evaluating online heuristic algorithms, various overbooking protection coefficients, and different scaling methods, a non-deterministic formulation was considered for optimizing four objective functions (energy consumption, economic savings, service quality, and resource utilization). The experimental results, considering 96 different scenarios, show that two of the online phase heuristic algorithms, taking into account a memetic algorithm for the offline phase, setting the overbooking protection coefficient to 0.75, and scaling the four objective functions based on the shortest Euclidean distance to the origin, yield the best performance.

1. INTRODUCTION

Given the increasing popularity of cloud computing among users, major companies providing cloud computing services, such as Google and Microsoft, have established vast data centers worldwide, with their energy consumption rising daily [1, 2]. A significant portion of the energy wastage in cloud data centers occurs in their hardware infrastructure, including servers, memory resources, and network equipment. Since idle hardware consumes a percentage of the power it uses during full operation, underutilizing these resources can lead to energy wastage. Therefore, low utilization rates of hardware resources are one of the reasons for energy wastage in data center infrastructures.

* Corresponding Author: m.h.mahmoodian@aut.ac.ir
Faculty of Electrical Engineering, Amirkabir University of Technology, Tehran, Iran

http://dx.doi.org/10.47176/TMI.2020.199

One of the most effective methods to reduce energy consumption during low-load periods in cloud data centers is to consolidate workloads onto the fewest possible physical resources and power down idle resources. Thus, this method employs live migration technology for virtual machines to consolidate virtual machines running on multiple physical servers with minimal workloads [3, 4].

Following the review of previous works and the classification of VMP issues, we will present the proposed method and formulate the problem. Finally, through the design and execution of various experiments, we will evaluate the proposed method and review the results obtained from this study.

2. LITERATURE REVIEW

The VMP issue in cloud computing has been extensively studied [5]. Initially, we focused on specific areas including: (1) techniques optimized for energy that have been applied to the problem [6, 7, 8]; (2) specific architectures like federated clouds where VMP issues are considered [9]; and (3) methods for comparing the performance of deployment algorithms in large demand-based clouds [10, 11].

Beloglazov et al. [7] reviewed energy-aware resource allocation policies and algorithms considering QoS. The identified challenges in energy-aware management for cloud data centers include: (1) developing fast energy-optimized algorithms for VMP problems considering multiple resources for large systems with predictive workload peaks to prevent performance degradation, (2) energy-aware optimization of virtual network topologies among VMs for optimal deployment to reduce network traffic and consequently the energy consumed by network infrastructure, (3) developing new temperature management algorithms for appropriate temperature control and energy consumption, (4) developing workload-aware resource allocation algorithms considering that current approaches assume uniform workloads, and (5) decentralization and distributed approaches to provide scalability and fault tolerance in solving the VMP problem.

Salimian et al. [12] reviewed various selection and deployment algorithms for efficient energy management in cloud data centers. The modeling approaches for virtual and physical resources, applied techniques, and future works for each studied article were identified. The most relevant future works include: (1) VMP for multi-core architectures considering multiple resources, (2) dynamic threshold considerations for QoS, and (3) developing intelligent schemes considering workload and live migration.

Gahlawat et al. [9] provided a brief review of the main architectures of federated clouds and the approaches considered for formulating the VMP problem. Federated cloud provides a platform for connecting the infrastructures of different cloud service providers, mainly aimed at responding to workload peaks.

Additionally, Mills et al. [10] compared methods for evaluating the performance of deployment algorithms in large demand-based clouds, considering 18 different VMP algorithms and 39 variables such as reallocation rate, user request rate, allocation rate, and disk space utilization.

The aforementioned reviews and research papers focus on specific aspects related to the VMP problem. Therefore, a review of related VMP issues is presented to identify research opportunities for a comprehensive overview of this research field.

In this section, we reviewed the conducted works. Generally, the criteria for classifying VMP issues in the reviewed works are presented in Figure 1. The most important point to note is that all these works have addressed the problem in a single phase.

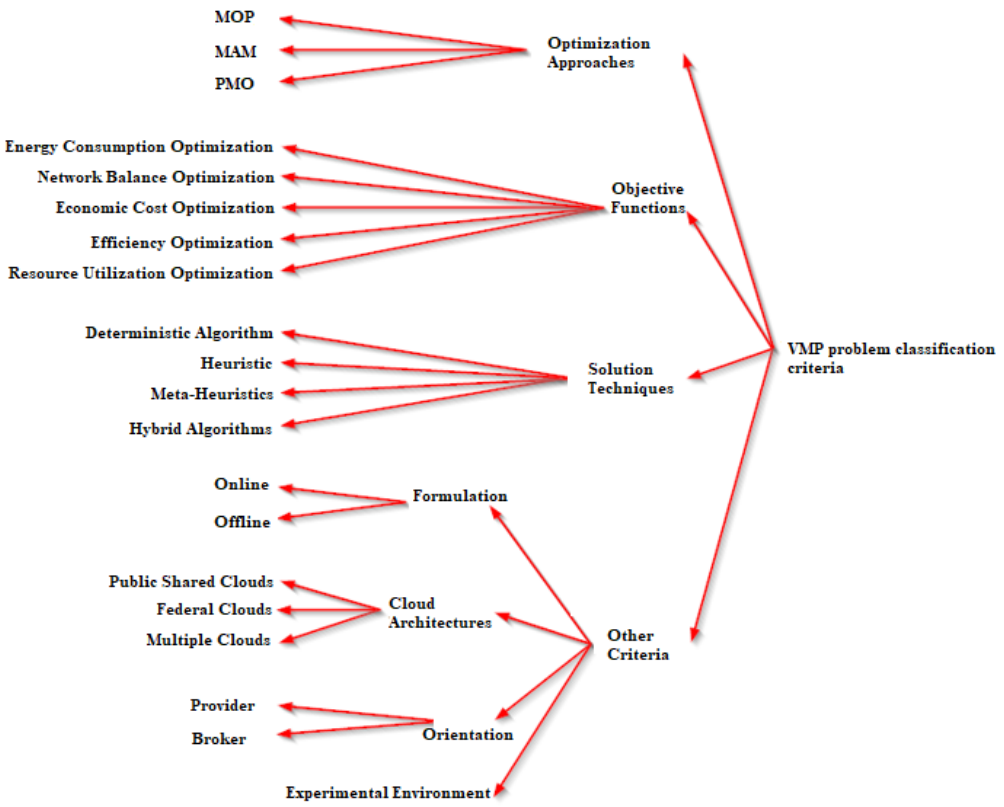

Fig.1. Summary of the Classification Criteria for VMP Issues Considered in Previous Works

Given that multiple objective functions and various approaches to modeling these functions exist, considering them simultaneously can improve the solution to VMP issues. This problem remains challenging and can be examined from various perspectives. Additionally, no method has yet been investigated for combining and simultaneously using online and offline algorithms to solve the VMP problem. All these points indicate that there is still much to explore in this research area.

3. PROPOSED METHOD

To enhance the quality of the solutions obtained by online algorithms, the VMP problem can be considered as a two-stage optimization problem where we simultaneously incorporate the advantages of online and offline formulations. For this purpose, the problem is considered to consist of two sub-problems: (1) Online VMP and (2) Offline VMP.

The online VMP sub-problem addresses dynamic requests where VMs created, modified, or terminated at runtime must be managed. Therefore, some heuristic algorithms can be useful in this context. The offline VMP sub-problem aims to improve the quality of solutions obtained in the online phase. The online phase is considered in all time intervals, while the offline phase can be pre-scheduled, for instance, periodically. When the offline phase begins, the placement of VMs is recalculated, hence this phase can be referred to as the reconfiguration phase, while the online phase continues to operate.

4. VMP PROBLEM FORMULATION

The set of physical machines belonging to an IaaS provider, as shown in Equation (1), is represented by the matrix H in $H \in R^{n\times(r+2)}$, where each physical machine Hi is characterized by r different physical resources. In this work,

we consider three different physical resources (Pr1-Pr3), which are: CPU[ECU] ،RAM Memory [GB], and Network Capacity [Mbps]. Additionally, the maximum energy consumption [W] is also considered. Finally, considering that an IaaS provider can have more than one cloud data center, a data center identifier is also included. Therefore, we have:

$$H_i = [Pr_{1,i} \quad \cdots \quad Pr_{r,i} \quad pmax_i \quad c_k] \quad \forall i \in \{1, \dots, n\} \tag{1}$$

The set of VMs requested by customers at any discrete time t, as shown in Equation (2), is represented by the matrix $V(t) \in \mathbb{R}^{m\times(r+3)}$. In this work, each virtual machine V_j requires three different virtual resources (Vr1-Vr3): CPU[ECU], RAM Memory [GB], Network Capacity [Mbps], $r = 3$. Additionally, economic savings and the SLA associated with each virtual machine are included. Therefore, we have:

$$V_j(t) = \left[Vr_{1,j}(t) \quad \cdots \quad Vr_{r,j}(t) \quad b_j(t) \quad R_j(t) \quad SLA_j(t)\right] \ \forall j \in \{1, \dots, m(t)\} \tag{2}$$

When a virtual machine V_j is turned off by the customer, its virtual resources are released, allowing IaaS providers to reuse these resources.

To model a dynamic VMP environment considering vertical and horizontal elasticity of cloud services, the set of requested VMs V(t) can include one of the following request types at any moment t: creating cloud services, scaling up/down VM resources, scaling up/down cloud services, and terminating cloud services for deploying cloud services at any moment t.

The resource utilization of a virtual machine V_j at any discrete time t is represented by the matrix $U(t) \in \mathbb{R}^{m(t)\times r}$and we have:

$$U_j(t) = [Ur_{1,j}(t) \quad \cdots \quad Ur_{r,j}(t)] \ \forall j \in \{1, \dots, m(t)\} \tag{3}$$

where $Ur_{k,j}(t)$is the utilization ratio of $Vr_k(t)$in the virtual machine V_j at any discrete time t.

The current placement of VMs on PMs x(t) represents the VMs requested at the previous discrete time (t-1); therefore, the dimensions of x(t) are based on the number of virtual machines m(t-1). Hence, the placement at any discrete time t is defined as the matrix x(t) $x(t) \in \{0,1\}^{n\times m(t-1)}$as shown in Equation (4):

$$x_j(t) = \left[x_{1,j}(t) \quad x_{2,j}(t) \quad \cdots \quad x_{n,j}(t)\right] \tag{4}$$

where $x_{i,j}(t) \in \{0,1\}$indicates whether V_j at discrete time t is allocated to the physical machine $x_{i,j}(t) = 1$or not $x_{i,j}(t) = 0$).

Online VMP:

In online algorithms for solving VMP issues, deployment decisions are made at each discrete time t. The online VMP problem formulation is based on [13], proposing that considering an IaaS environment consisting of a set of PMs (H), a set of active VMs requested before time t (V(t)), and the current deployment of VMs on PMs x(t), we aim to deploy V(t) on H for the discrete time t+1 to form x(t+1) without migrating x(t), considering the satisfaction of the problem constraints and the optimization of the objective functions.

Therefore, the deployment at time t+1 is represented by the matrix $x(t+1) \in \{0,1\}^{n\times m(t)}$ and we have:

$$x_j(t+1) = \left[x_{1,j}(t+1) \quad x_{2,j}(t+1) \quad \cdots \quad x_{n,j}(t+1)\right] \tag{5}$$

Offline VMP:

An offline algorithm solves the VMP problem through VM migration among PMs considering a static (non-dynamic) environment where VM requests do not change over time. The offline VMP problem formulation, based on [9], is proposed such that considering the current deployment of VMs on PMs (x(t)), we aim for a reconfiguration through VM migration among PMs for discrete time $x'(t)$, ensuring that constraints are satisfied and the objective functions are optimized.

The result of the offline VMP problem is the reconfiguration of deployment through VM migration among PMs for discrete time $x'(t)$, represented by the reconfiguration x(t) and indicating the output of the offline VMP process.

Constraints:

Constraint 1: Unique Deployment of Virtual Machines:

A virtual machine V_j must be deployed on a single physical machine H_i for execution. Any V_j with an SLA value less than the maximum possible value (i.e., $SLA_j = s$) may not be deployed on any PMs:

$$\sum_{i=1}^{n} x_{j,i}(t) \leq 1 \qquad \forall j \in \{1, \cdots, m(t)\}, \text{for all VM } V_j \tag{6}$$

Constraint 2: Ensuring SLA Compliance:

A virtual machine V_j with the highest SLA level (i.e., $SLA_j = s$) must be forcibly allocated to a physical machine Hi for execution. Thus, this constraint is mathematically expressed as:

$$\sum_{i=1}^{n} x_{j,i}(t) = 1 \qquad SLA_j = s \quad \text{so that} \quad \forall j \tag{7}$$

Constraint 3: Physical Resources of Physical Machines:

A physical machine Hi must have sufficient resources to meet the dynamic requirements of all virtual machines V_j allocated to it for execution. It is important to remember that VM resources are used dynamically to allow the reuse of idle reserved resources. Sometimes, reusing idle resources can lead to unmet demands. Therefore, providers need to reserve a percentage of idle resources as protection, defined by the overbooking protection coefficient λ_k, which adjusts the number of reserved resources to reduce SLA violations:

$$\sum_{j=1}^{m(t)} \left\{ Vr_{k,j} \times Ur_{k,j} + \left[Vr_{k,j} \times \left(1 - Ur_{k,j}\right) \right] \times \lambda_k \right\} \times x_{j,i}(t) \leq Pr_{k,i} \tag{8}$$

$\forall i \in \{1, \cdots, n\}$ and $\forall k \in \{1, \cdots, r\}$ i.e., for each physical machine Hi text and each resource considered r where $\lambda_k$is the protection coefficient for $Vr_{k,j} \in [0,1]$.

Objective Functions:

1. Minimizing Energy Consumption:

This is modeled as follows:

$$\min f_1(x,t) = \sum_{i=1}^{n} ((pmax_i - pmin_i) \times Ur_{1,i}(t) + pmin_i) \times Y_i(t) \tag{9}$$

2. Maximizing Economic Savings:

A provider should offer its idle resources to the cloud community at a price lower than the actual cloud market. The pricing can depend on a special agreement between the cloud community providers. Consequently, we assume that the main provider rents the requested resources that cannot be provided at 70% $\hat{X}_j$ of the market price R_j. This objective is formulated as:

$$LC = \sum_{j=1}^{m(t)} (R_j \times X_j \times \hat{X}_j) \tag{10}$$

Additionally, overbooked resources may sometimes fail to meet demand during certain periods, leading to QoS degradation, SLA violations, and financial penalties. Therefore, we consider the following relationship for economic penalties:

$$EP = \sum_{j=1}^{m(t)} (\sum_{k=1}^{r} Rr_{k,j}(t) \times \Delta r_{k,j}(t) \times X_j(t) \times \emptyset) \tag{11}$$

Thus, we have:

$$\min f_2(x,t) = LC + EP \tag{12}$$

3. Maximizing Service Quality:

This objective proposes maximizing service quality by placing the maximum number of VMs with the highest SLA priority over VMs with lower SLA. To evaluate this objective in the context of minimization, the total SLA violation is minimized and formulated as follows:

$$\min f_2(x,t) = LC + EP \tag{13}$$

4. Maximizing Resource Utilization:

Efficient resource utilization is a managerial challenge for IaaS providers. This objective formulates the strategy of maximizing resource consumption by minimizing the average coefficient of wasted resources for each physical machine Hi (i.e., resources not allocated to any virtual machine V_j. his objective function is formulated as follows:

$$\min f_4(x,t) = \frac{\sum_{i=1}^{n} \left[1 - \left(\frac{Ur_{1,i}(t) + \cdots + Ur_{r,i}(t)}{r}\right)\right] \times Y_i(t)}{\sum_{i=1}^{n} Y_i(t)} \tag{14}$$

Scaling and Normalization Methods:

To make the cost of each objective function comparable and combinable as a single objective, they must be normalized. In this work, the cost of each objective function is normalized by computing the following relationship, where $\hat{f}_i(x,t) \in \mathbb{R}$ and $0 \leq \hat{f}_i(x,t) \leq 1$:

$$\hat{f}_i(x,t) = \frac{f_i(x,t) - f_i(x,t)_{min}}{f_i(x,t)_{max} - f_i(x,t)_{min}} \tag{15}$$

This work examines multiple methods that can be applied to a multi-objective optimization problem to convert the normalized objective functions into a comparable value. The three scaling methods evaluated are: weighted sum, Euclidean distance, and Chebyshev distance, presented in relationships (16) to (18), respectively.

$$\min F(x,t) = \sum_{i=1}^{q} \hat{f}_i(x,t) \times w_i \tag{16}$$

$$\min F(x,t) = \sqrt{\sum_{i=1}^{q} \left|\hat{f}_i(x,t)\right|^2} \tag{17}$$

$$\min F(x,t) = \sum_{i=1}^{q} \left|\hat{f}_i(x,t)\right| \tag{18}$$

Let F(x,t) be a combined value of $\hat{f}_i(x,t)$. Here, wi represents the weight signifying the importance related to $f_i(x,t)$, which is considered constant and equal to 0.25 in this work. Moreover, q indicates the number of objective functions, which in our case is 4.

For each scenario $s \in S$ the instantaneous average of the objective function F(x,t) given in equations (16) to (18) is computed as follows:

$$\overline{f_s(x,t)} = \frac{\sum_{t=1}^{t_{max}} F(x,t)}{t_{max}} \tag{19}$$

When parameters are not deterministic, solutions acceptable for all (or most) scenarios considered $s \in S$ must be found. In this work, we consider the mean, maximum, and minimum criteria for selection, as given in equations (20) to (22):

$$F_1 = \overline{F(x,t)} = \frac{\sum_{s=1}^{|S|} \overline{f_s(x,t)}}{|S|} \tag{20}$$

$$F_2 = \max_{s \in S}\left(\overline{f_s(x,t)}\right) \tag{21}$$

$$F_3 = \min_{s \in S}\left(\overline{f_s(x,t)}\right) \tag{22}$$

where F1 is the mean, F2 is the maximum, and F3 is the minimum $\overline{f_s(x,t)}$ for all scenarios $s \in S$.

Suitable Algorithms for the Online Phase:

To analyze the formulated VMP problem in the online phase, we employed the heuristic algorithms First Fit (FF), Best Fit (BF), Worst Fit (WF), First Fit Decreasing (FFD), and Best Fit Decreasing (BFD).

Memetic Algorithm for the Offline Phase:

To analyze the formulated VMP problem in a static manner, many algorithms have been proposed. After various evaluations, we chose the memetic algorithm due to its special attention to the system reconfiguration time and its applicability to real-world scales (typically involving thousands of VMs and PMs). In reality, the VMP problem is dynamic, and algorithms proposed for solving the static VMP problem often fall short both in practical application and in focusing on reconfiguration time. Fortunately, the memetic algorithm, besides being scalable, has suitable speed, allowing its use in the offline phase, where time is crucial.

Thus, in this work, a memetic algorithm for solving the offline formulated VMP problem with multiple objectives is considered. This algorithm is based on the algorithm proposed by Ihara et al. in [14], and its pseudocode is shown in Figure 2:

```
Data: H, V(t), U(t), x(t)
Result: Incremental Placement x′(t)
initialize set of candidate solutions Pop0;
Pop′0 = repair infeasible solutions of Pop0;
Pop″0 = apply local search to solutions of Pop′0;
x′(t) = select best solution from Pop″0 ∪ x(t) considering equations (16 to 18);
process scale-up of VMs resources from V (t);
u = 0; Popu = Pop″0;
while stopping criterion is not satisfied do
    Popu    = selection of solutions from Popu ∪ x′(t);
    Pop′u   = crossover and mutation on solutions of Popu;
    Pop″u   = repair infeasible solutions of Pop′u;
    Pop‴u   = apply local search to solutions of Pop″u;
    x′(t)   = select best solution from Pop‴u considering equations (16 to 18);
    increment number of generations u;
End
return x′(t)
```

Fig.2. Pseudocode of the Memetic Algorithm (MA).

5. EXPERIMENTAL ENVIRONMENT AND RESULTS

The evaluated algorithms introduced in Chapter 4 have been implemented using the Java programming language. Experiments were run on a Windows 10 operating system with an Intel Core i5-3210M CPU at 2.5GHz and 16GB of RAM. The number of resources considered for our problem is r = 3. The protection coefficient for each resource was initially set to $\lambda_k = 0.5$.The penalty coefficient for each resource was set to $\emptyset = 1$.

Moreover, to prioritize services with higher SLA over those with lower SLA, the parameter $\hat{C}$ was set to 1000, assuming the highest priority in VMs is marked by s = 4. Additionally, the physical resources (matrix H) represent a heterogeneous IaaS cloud, considering four types of PMs. As input data, the physical machines are one of the four different types provided in Table 1. Considering the available types of physical machines, we consider two types of scenarios as given in Table 2.

Table 1. Specifications of Resources for Different Types of Physical Machines Considered

Sources / PM Type	CPU [ECU]	RAM [GB]	Networking [Mbps]	pmax [W]
Small (S)	32	128	1000	800
Medium (M)	64	256	1000	1000
Large (L)	256	512	1000	3000
Extra Large (XL)	512	1024	20000	5000

Table 2. Details of the Number of Each Type of Physical Machine Considered for Each Scenario

PM Type	Quantity in Scenario 1	Quantity in Scenario 2
Small (S)	50	20
Medium (M)	50	20
Large (L)	50	15
Extra Large (XL)	30	8

Additionally, 48 different workload sequences of requested cloud services (V(t)) and their characteristics, along with their resource utilization (U(t)), are also considered as input. The requested virtual machines are based on Amazon EC2's proposed samples.

Finally, for evaluation, we have designed four experiments: Experiment 1: Evaluation of the proposed two-stage algorithm compared to single-stage algorithms, Experiment 2: Online algorithms for the online VMP phase,

Experiment 3: Evaluation of overbooking protection coefficients, and Experiment 4: Evaluation of scaling methods. The results of each of these experiments are presented in Tables 3 to 6, respectively.

Table 3. Experiment 1: Average (F_1), Maximum (F_2), and Minimum (F_3) Costs of Objective Functions for Each of the Three Algorithms Considering the Protection Coefficient (λ_k) as 0.5 and Euclidean Distance to the Origin (ED) as the Scaling Method. Best results are highlighted.

Protection Coefficient (λk)	Scaling Method	Criterion	Algorithm No.	Online VMP	Offline VMP	Scenario 1	Scenario 2	Mean	Rank
0.50	ED	F_1	1	BFD	-	0.697	0.858	0.778	3rd
			2	-	MA	**0.601**	**0.767**	**0.684**	**1st**
			3	BFD	MA	0.636	0.819	0.728	2nd
		F_2	1	BFD	-	0.778	0.921	0.850	3rd
			2	-	MA	**0.698**	**0.834**	**0.766**	**1st**
			3	BFD	MA	0.737	0.872	0.805	2nd
		F_3	1	BFD	-	0.611	0.690	0.651	3rd
			2	-	MA	**0.488**	**0.641**	**0.565**	**1st**
			3	BFD	MA	0.532	0.673	0.603	2nd

Table 4. Experiment 2: Considering the Average Criterion for the ith Objective Function Cost F_1^i to Evaluate the Heuristic Algorithms with the Protection Coefficient (λ_k) as 0.5 and Euclidean Distance to the Origin (ED) as the Scaling Method. Best results are highlighted.

Protection Coefficient (λk)	Scaling Method	Algorithm	F_1^i	Scenario 1	Scenario 2	Average of Scenarios	Mean	Rank
0.50	ED	FF	F_1^1	0.674	0.885	0.780	0.790	4th
			F_1^2	0.493	0.785	0.639		
			F_1^3	0.857	0.988	0.923		
			F_1^4	0.768	0.870	0.819		
		BF	F_1^1	**0.548**	0.812	0.680	0.731	2nd
			F_1^2	**0.430**	0.723	**0.577**		
			F_1^3	**0.787**	0.983	0.885		
			F_1^4	**0.707**	0.858	**0.783**		
		WF	F_1^1	0.685	0.920	0.803	0.806	5th
			F_1^2	0.502	0.792	0.647		
			F_1^3	0.857	0.991	0.924		
			F_1^4	0.800	0.901	0.851		
		FFD	F_1^1	0.599	0.869	0.734	0.769	3rd
			F_1^2	0.482	0.772	0.627		
			F_1^3	0.832	**0.979**	0.906		
			F_1^4	0.755	0.860	0.808		
		BFD	F_1^1	0.577	**0.798**	0.688	**0.826**	1st
			F_1^2	0.459	**0.689**	**0.574**		
			F_1^3	0.801	**0.979**	0.890		
			F_1^4	**0.707**	**0.810**	**0.759**		

Table 5. Experiment 3: Considering the Average Criterion (F_1) for Objective Function Costs to Evaluate the Two Best Performing Heuristic Algorithms from Experiment 2, with Various Protection Coefficients (λ_k) and Euclidean Distance to the Origin (ED) as the Scaling Method. Best results are highlighted.

Criterion	Scaling Method	Algorithm	Protection Coefficient (λ_k)	Scenario 1	Scenario 2	Average of Scenarios
F_1	ED	BF	0.00	0.612	0.856	0.734
			0.25	0.621	0.852	0.737
			0.50	0.618	0.844	0.731
			0.75	0.619	0.844	0.732
			1.00	0.621	0.843	0.732
		BFD	0.00	0.632	0.827	0.730
			0.25	0.640	0.823	0.732
			0.50	0.636	0.819	0.728
			0.75	0.632	0.816	0.724
			1.00	0.632	0.818	0.725

Table 6. Experiment 4: Considering the Average Criterion for the i-th Objective Function Cost F_1^i to Evaluate the Heuristic Algorithm BFD with Protection Coefficient (λ_k) as 0.75 and Various Scaling Methods. Best results are highlighted.

Algorithm	Protection Coefficient (λ_k)	F_1^i	Scaling Method	Scenario 1	Scenario 2
BFD	0.75	F_1^1	WS	0.578	0.812
			ED	0.573	0.795
			CD	0.589	0.825
		F_1^2	WS	0.453	0.680
			ED	0.456	0.682
			CD	0.454	0.676
		F_1^3	WS	0.805	0.983
			ED	0.799	0.978
			CD	0.814	0.989
		F_1^4	WS	0.712	0.802
			ED	0.698	0.808
			CD	0.702	0.814

From Experiment 1, we understand that the proposed two-stage algorithm is more practical compared to single-stage offline algorithms because it considers the dynamic nature of the environment and yields better results than single-stage online algorithms due to the reconfiguration capability improving the solutions. From Experiment 2, we find that the best algorithms for the online phase of the VMP problem are BFD and BF, respectively. In the offline phase, we used the MA algorithm. Experiment 3 aimed to determine the best value for overbooking protection coefficients in each of the two scenarios based on the results from the previous two experiments. Finally, in Experiment 4, three main scaling methods (ED, WS, and CD) were evaluated based on the results from the previous three experiments, with Euclidean Distance to the origin identified as the most suitable scaling method.

6. CONCLUSION

In this study, we focused on two-phase optimization considering a more realistic representation of cloud infrastructures and more complex VMP environments with overbooking considerations and elasticity capabilities. Additionally, experiments were conducted on real workload effects considering 96 different scenarios. This work

presented the first experimental comparison of five different heuristic algorithms for solving the VMP problem in the online phase to optimize four objectives.

Moreover, besides the experiments to evaluate the proposed two-phase algorithm and compare heuristic algorithms for the online phase of the VMP problem, overbooking protection coefficients and scaling methods were also examined. Given the randomness of customer requests, VMP problem-solving algorithms under uncertainty conditions were evaluated, considering various uncertainty parameters.

Finally, with all the analyses conducted, we can claim that the proposed method is capable of solving the VMP problem under any workload and scenario and is practically applicable.

Transparency Statement

The data supporting this study are available upon reasonable request to the corresponding author, subject to ethical and confidentiality considerations.

Acknowledgments

We would like to express our gratitude to all individuals who contributed to this project.

Declaration of Interest

The authors declare that they have no competing interests.

Funding

This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

REFERENCES

[1] Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications, 52*, 11-25. https://doi.org/10.1016/j.jnca.2015.02.002

[2] Zhang, F., Liu, G., Fu, X., & Yahyapour, R. (2018). A survey on virtual machine migration: Challenges, techniques, and open issues. *IEEE Communications Surveys & Tutorials, 20*(2), 1206-1243. https://doi.org/10.1109/COMST.2018.2794881

[3] Silva Filho, M. C., Monteiro, C. C., Inácio, P. R., & Freire, M. M. (2018). Approaches for optimizing virtual machine placement and migration in cloud environments: A survey. *Journal of Parallel and Distributed Computing, 111*, 222-250. https://doi.org/10.1016/j.jpdc.2017.08.010

[4] Prodan, R., et al. (2019). Dynamic multi-objective virtual machine placement in cloud data centers. In *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)* (pp. 92-99). Kallithea-Chalkidiki, Greece. https://doi.org/10.1109/SEAA.2019.00023

[5] Talebian, H., Gani, A., Sookhak, M., Abdelatif, A. A., Yousafzai, A., Vasilakos, A. V., & Yu, F. R. (2019). Optimizing virtual machine placement in IaaS data centers: Taxonomy, review and open issues. *Cluster Computing, 22*(4), 1527-1568. https://doi.org/10.1007/s10586-019-02954-w

[6] Malmodin, J., & Lundén, D. (2018). The energy and carbon footprint of the global ICT and E&M sectors 2010-2015. *Sustainability, 10*(9), 3027. https://doi.org/10.3390/su10093027

[7] Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems, 28*(5), 755-768.

https://doi.org/10.1016/j.future.2011.04.017

[8] Xiao, H., Hu, Z., & Li, K. (2019). Multi-objective VM consolidation based on thresholds and ant colony system in cloud computing. *IEEE Access, 7*, 53441-53453. https://doi.org/10.1109/ACCESS.2019.2912722

[9] Gahlawat, M., & Sharma, P. (2014). Survey of virtual machine placement in federated clouds. In *2014 IEEE International Advance Computing Conference (IACC)* (pp. 735-738). IEEE. https://doi.org/10.1109/IAdCC.2014.6779415

[10] Mills, K., Filliben, J., & Dabrowski, C. (2011). Comparing VM-placement algorithms for on-demand clouds. In *2011 IEEE Third International Conference on Cloud Computing Technology and Science* (pp. 91-98). IEEE. https://doi.org/10.1109/CloudCom.2011.22

[11] Saadi, Y., & El Kafhali, S. (2020). Energy-efficient strategy for virtual machine consolidation in cloud environment. *Soft Computing, 24*(9), 6935-6954. https://doi.org/10.1007/s00500-019-04312-2

[12] Salimian, L., & Safi, F. (2013). Survey of energy efficient data centers in cloud computing. In *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing* (pp. 369-374). IEEE Computer Society. https://doi.org/10.1109/UCC.2013.81

[13] Li, M., Bi, J., & Li, Z. (2016). Improving consolidation of virtual machine based on virtual switching overhead estimation. *Journal of Network and Computer Applications, 59*, 158-167. https://doi.org/10.1016/j.jnca.2015.07.008

[14] Ihara, D., López-Pires, F., & Baran, B. (2015). Many-objective virtual machine placement for dynamic environments. In *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)* (pp. 75-79). IEEE. https://doi.org/10.1109/UCC.2015.22