



Multi Objective Decision Making for Virtual Machine Placement in Cloud Computing

Wissal Attaoui¹(✉), Essaid Sabir^{1,2}(✉), Halima Elbiaze²(✉),
and Mohamed Sadik¹(✉)

¹ NEST Research Group, ENSEM, Hassan II University of Casablanca,
Casablanca, Morocco

{w.attaoui,e.sabir,m.sadik}@ensem.ac.ma

² Computer Science Department, University of Quebec at Montreal (UQAM),
Montreal, QC H2L 2C4, Canada
elbiaze.halima@uqam.ca

Abstract. Cloud computing is an innovative process that delivers on-demand services over the internet. Virtualization is considered as the key concept of cloud computing since it handles running multiple virtual resources in a single physical resource. Mapping the virtual machine (VM) to the appropriate physical machine (PM) is called virtual machine placement (VMP). In this context, the dilemma of placing VMs in the cloud environment presents a significant challenge that has been wholly addressed but not yet totally fixed. This paper provides a multi-objective decision-making approach for VMP in a cloud computing infrastructure. We propose a conic scalarization method to solve the optimization problem. Simulation results prove that the offline algorithm yields good results compared to online deterministic algorithms.

Keywords: Cloud computing · Virtual machine placement

1 Introduction

Nowadays, cloud computing is considered as an emerging technology that regularly evolves towards a significant field of computer science. Cloud services are continually being expanded to meet customer demands where the Infrastructure as a Service (IaaS) package is the most requested by cloud users. The cloud service provider delivers hardware services (i.e., CPU, memory, storage, network bandwidth, etc.) using virtualization technology. To execute or serve a task in the cloud, the first process to be performed is the Virtual Machine (VM) allocation, which is the process of allocating or mapping a VM with a specific configuration where the assigned VM must meet the Quality of Service (QoS). The next step is the VM placement process, which is the process of placing or mapping the VM to its best fit physical machine (PM), as presented in Fig. 1. By analogy,

we can see this problem as a “Tetris” game whose goal is to place the arrived shapes in the right places properly. From the provider’s point of view, energy consumption, cost, and resource wastage are the main objectives that need to be optimized. At the same time, the QoS and the quality of experience are the critical elements to be maximized from the customer perspective. In this context, many algorithms and policies were proposed to solve the VM allocation and placement problems.

The traditional VM mapping approaches were treated as single-objective, whereas the recent ones address the VMP as a multi-objective optimization problem. Moreover, this placement process can be accomplished either offline (static) or online (dynamic). On the one hand, for offline placement, the data center (DC) providers consolidate data and make placement decisions to meet the consumer requests considering different constraints. On the other hand, for online placement, the DC suppliers gather data periodically then decide whether a rescheduling of the placement procedure is required.

Virtualization gives the possibility to conveniently move a Virtual Machine (VM) from a specific host to another without turning it off; therefore, this can offer a dynamicity on VM placement optimization with a negligible impact on performance. Despite its numerous benefits, these dynamics may result in sub-optimal or unstable configurations of the virtual networks. Moreover, VMs may experience some fluctuations within the resource utilization (e.g., a mobile application server and a web server may possess identical patterns of the incoming workload while using the same CPU). In this context, several challenges hinder the efficient placement of the virtual machine, which can be dealt with as multi-objective optimization approaches; Considering various trade-offs between energy consumption, reliability and performance degradation [7, 10], power consumption and resource utilization [13], cost and QoS [1, 21], network traffic and resource utilization [6, 8, 18], etc.

Accordingly, to handle these combinatorial optimization problems, we model the VM placement problem as a multi-objective decision making (MODM) approach aiming to simultaneously optimize five objective functions: energy consumption, cost, network traffic, resource utilization, and QoS. Therefore, to solve this optimization model, we transform the problem into a single objective function using a scalarization method.

This paper is structured as follows; the next section presents the related works. Section 3 presents the problem formulation analyzed as a multi-objective decision making (MODM) approach. In Sect. 4, we describe the scalarization method proposed to solve the MODM problem. Section 5 presents the test environment to evaluate our model and examine the most appropriate algorithm among proposals.

2 Related Works

Several approaches in the literature have studied the VMP problem in both static and dynamic environments. In the following, we cite some references that

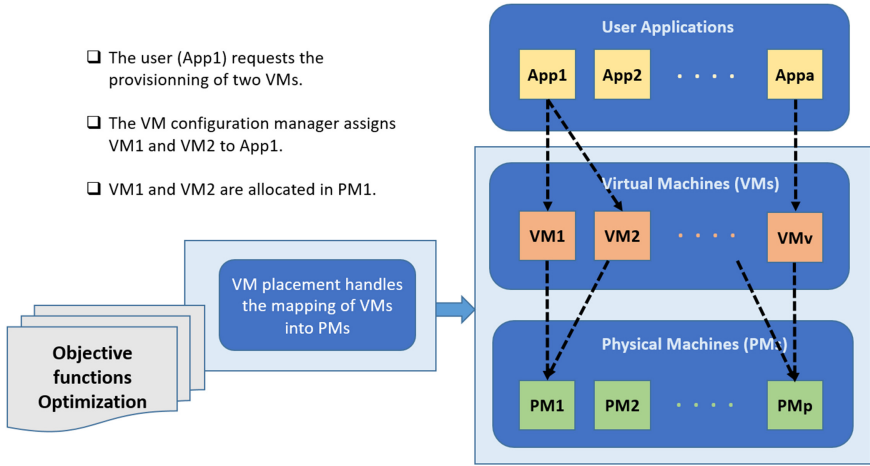


Fig. 1. Cloud computing architecture

have dealt with this placement problem in different ways, mono-objective or multi-objective optimization in online or offline settings. From 2007 to 2030, global energy consumption will increase by 76%, referring to *energy outlook 2030*. Therefore, we should think about the methods to be followed to reduce the energy consumption in the cloud; since cloud data centers included thousand of servers that consume an enormous amount of power. Therefore, to save energy consumption, it's preferable to place VMs on as few PMs as possible. However, if VMs are placed densely in a server, this can cause an occurrence of heat islands, which can affect the reliability of the device. In this context, [7] proposes a bi-objective optimization algorithm that considers energy-saving and server reliability. This algorithm aims to minimize the total power consumption of all servers as a function of resource utilization and simultaneously achieves reliability by adding backup servers when the number of working servers exceeds a redundancy ratio. Simulation results prove the performance of the recommended algorithm in terms of power consumption in both working and backup servers compared to online deterministic heuristics as First-Fit-Decreasing (FFD), Modified FFD (MFFD) and Thermal Aware Workload Scheduling Algorithm (TASA).

In the same regard, unbalanced resource usage may lead to resource wastage, SLA violation, and high power consumption. Authors in [13] propose a multi-objective virtual machine placement algorithm for IaaS cloud named (MOVMP). This model allows reducing the number of active PMs through migrating VMs and achieving a more balanced use of resources, which minimizes the energy consumption and resource wastage. The experiment setup of the proposed algorithm yields good results compared to First-Fit (FF) and Power-Aware Best Fit Decreasing (PABFD).

The quality of service (QoS) in cloud computing is another challenge for cloud providers to attract and satisfy users. Authors in [1] treated the VMP problem

as a tradeoff between QoS and cost. They propose a two-layer model, where the first one considers the allocation problem in a cost-effective way using linear programming for load balancing. The second one regards the VM placement by proposing a Genetic Algorithm Based Virtual Machine Placement (GABVMP) algorithm as a solution to the optimization problem. Simulation results show that GABVMP is more performant than two greedy heuristics (Random placement and FF placement).

Several papers study the problem of minimizing network traffic [6, 8, 18] to enhance the performance of a DC by selecting the most suitable physical machines for virtual machines. Daniel et al. [9] present a VMP algorithm to reallocate virtual machines in DC Server contingent on the memory usage, the traffic matrix network, and the overall CPU. The first phase of this VMP algorithm considers collecting data from VMs and DC topology. The second one focuses on partitioning servers with a higher level of connectivity. The last one consists of clustering VMs by defining the amount of purchased traffic using graph theory to manage all the virtual servers. Simulation results prove that this solution improves network traffic quality and the availability of bandwidth at DC.

3 Multi-objective Decision Making: Problem Formulation

Based on our literature survey in [3], we analyze the different problems that may interrupt the VM placement. We classify the existing solutions into five primary objective functions based on multiple performance metrics such as energy consumption, quality of service, service level agreement, resource usage, and incurred cost.

In this paper, we form the problem of virtual machine placement as a multi-objective decision making (MODM) approach, by optimizing the five objective functions: (1) Energy consumption minimization, (2) cost optimization, (3) Network traffic minimization, (4) Resource utilization and (5) Quality of service maximization simultaneously. We consider the available PMs specifications, the requested VMs, the network traffic between VMs and their current placement as Inputs, and the new convenient placement as Output. This section presents a theoretical approach to solve the VMP problem (see Fig. 2).

Given a set of physical machines $S = S_1, S_2, \dots, S_n$ and a set of virtual machines $V = V_1, V_2, \dots, V_m$. We are looking for new placement of VMs V on a set of PMs S while satisfying the constraints and simultaneously optimizing the five objective functions cited above in a pure multi-objective context.

Each physical machine S_i is characterized by a specific processing resources CPU, RAM memory, storage, and maximum power consumption represented as:

$$S_i = [S_i^{cpu}, S_i^{ram}, S_i^{stor}, S_i^{pmmax}]; \quad \forall i \in \{1, 2, \dots, n\} \quad (1)$$

Each virtual machine V_j needs processing resources of CPU V_j^{cpu} , RAM V_j^{ram} , and storage V_j^{stor} , providing economic revenue R_j and attributing SLA to each VM. Consequently, a virtual machine V_j will be represented as:

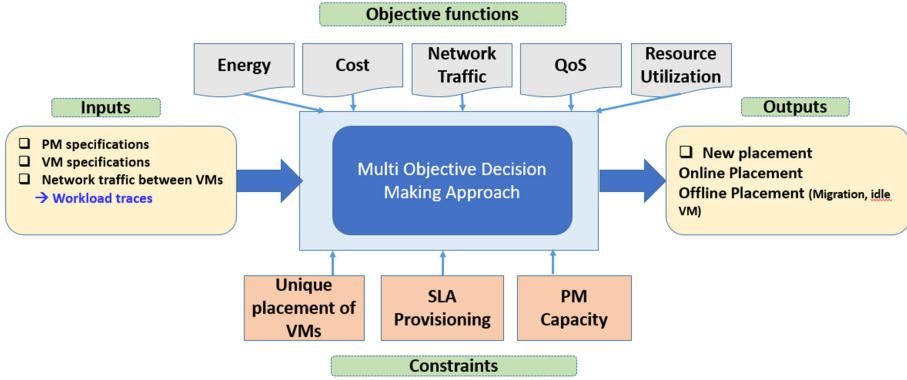


Fig. 2. Multi attribute decision making approach

$$V_j = [V_j^{cpu}, V_j^{ram}, V_j^{stor}, R_j, SLA_j]; \quad \forall j \in \{1, 2, \dots, m\} \quad (2)$$

The network traffic between the requested VMs is represented as follows:

$$T_j = [T_{j_1}, T_{j_2}, \dots, T_{j_m}]; \quad \forall j \in \{1, \dots, m\} \quad (3)$$

where T_{j_k} represents the average communication rate in [Kbps], between the virtual machine V_j and the virtual machine V_k . Note that we can consider $T_{jj} = 0$.

3.1 Objective Functions

The five objective functions are mathematically formulated as follows:

Energy Consumption Minimization

Based on [4], the energy consumption of a resource is defined as the sum of the power consumption of each PM considering a linear relationship with the CPU utilization:

$$\min g_1(x) = \sum_{i=1}^n [(P_i^{max} - P_i^{min}) \cdot U_i^{cpu} + P_i^{min}] \cdot A_i \quad (4)$$

where

- $g_1(x)$: the total energy consumption of the PMs;
- P_i^{max} : is the power consumption at the peak load;
- P_i^{min} : is the minimum power consumption in the inactive mode;
- U_i^{cpu} : is the utilization ratio of CPU resources used by S_i
- A_i : is a binary parameter equal to 1 if PM turned on, 0 otherwise.

Cost Optimization

Cloud service providers avoid economic loss generated by failures. Therefore, based on [22], we adopt the proposed approach of maximizing cloud provider's revenue regarding SLA violation cost. We have two cases:

- if V_j is finished without failure: provider will receive revenue:

$$R_j = c_j \cdot t_j - E_j \quad (5)$$

where c_j is the price of V_j ; E_j is the execution cost of V_j that can be estimated through paying site infrastructure Capex and energy costs overall machines, and t_j is the lifetime of VM V_j .

- if V_j is failed: the customer is eligible to receive a refund from the provider, which is equal to $D_j = c_j \cdot t_j \cdot p_{SLA}$, where p_{SLA} is a constant parameter and indicates the part of the entire invoice that the provider has to deliver.

The expected revenue generated from placing V_j on S_i is defined as the following:

$$F_j = R_j(1 - f_j^i) - (E_j + D_j) \cdot f_j^i, \quad (6)$$

where f_j^i is the probability of at least one failure occurring on S_i in t_j .

The problem of maximizing cloud provider's revenue (minimizing total cost) is expressed by:

$$\min g_2(x) = \sum_{j=1}^n \widehat{F}_j \cdot B_j; \quad (7)$$

where:

- $g_2(x)$: is the total economical revenue for placing VMs; $\widehat{F}_j = -F_j$
- B_j : is a binary variable equals to 1 if V_j is located on PM, 0 otherwise.

Network Traffic Minimization

Based on [16], the total network traffic among VMs is defined as the sum of average network traffic T_{jk} generated by each VM V_j , which is located to run on any PM, with other VMs V_k that are located to run on different PMs.

$$\min g_3(x) = \sum_{j=1}^n \sum_{k=1}^n (T_{jk} \cdot X_{jk}) \quad (8)$$

where:

- $g_3(x)$: is the total network traffic between VMs;
- X_{jk} : A binary variable equals to 1 if V_j and V_k are located in different PMs and 0 otherwise.

The traffic between two VMs V_j and V_k which are located on the same PM do not contribute to increase the total network traffic given by Eq. (8).

Resource Utilization

To make effective use of the resources in all dimensions and balance the resource utilization on each server along different dimensions, we adopt the resource wastage model proposed in [15]. The unused resources available on each server may change considerably with different VM placement solutions. The following equation is adopted to compute the implied cost of wasted resources:

$$W_i = \frac{|Y_i^p - Y_i^m| + \epsilon}{Z_j^p + Z_j^m} \quad (9)$$

where:

- W_i : denotes the resource wastage of the i -th server;
- Y_i^p and Y_i^m represent the normalized CPU and memory resource usage (i.e., the ratio of used resource to total resource);
- Z_j^p and Z_j^m represent the normalized remaining CPU and memory;
- ϵ is a very small positive real number and its value is set to be 0.0001;

The related objective function of resource wastage can be expressed by:

$$\min g_4(x) = \sum_{i=1}^m W_i \quad (10)$$

Quality of Service Maximization

Quality of Service maximization is achieved when locating maximum number of VMs with highest SLA level of priority. We use the same equation proposed in [16] aiming to minimize SLA violations by using the highest level of priority SLA_i :

$$\min g_5(x) = \sum_{j=1}^n C^{SLA_j} \cdot SLA_j \cdot B_j. \quad (11)$$

where:

- $g_5(x)$: is the total QoS function for a given placement;
- C : is a constant, sufficiently large, to give priority to services with a higher SLA over those with a lower SLA. Otherwise, if the constant C had small value, $g_5(x)$ might choose a large number of VMs with a lower priority, which is not correct, considering the intended purpose in Eq. (11).

3.2 Constraints

PMs Capacity. A PM must have sufficient available resources to meet the requirements of all VMs. The capacity constraints can be mathematically formulated as:

$$h_1(x) : \sum_{j=1}^m V_j^{cpu} \cdot P_{ji} - S_i^{cpu} \leq 0 \quad (12)$$

$$h_2(x) : \sum_{j=1}^m V_j^{ram} \cdot P_{ji} - S_i^{ram} \leq 0 \quad (13)$$

$$h_3(x) : \sum_{j=1}^m V_j^{stor} \cdot P_{ji} - S_i^{stor} \leq 0 \quad (14)$$

SLA Provisioning. A virtual machine V_j with decisive SLA (i.e., $SLA_j = 1$) must certainly be located to run on a physical machine S_i . Consequently, this restriction is expressed as:

$$h_4(x) : \sum_{i=1}^n S_{ji} = 1 \quad \forall j \text{ such that } SLA_j = 1 \quad (15)$$

Where:

- SLA_j : Service Level Agreement $SLA_j = 1$ if V_j is critical, or 0 otherwise.

Unique Placement of VMs. A VM V_j should be located to run on a single PM. Alternatively, it could not be located in any PM if the associated SLA_j is not the highest level of priority. Consequently, this constraint is expressed as:

$$h_5(x) : \sum_{i=1}^n S_{ji} - 1 \leq 0 \quad (16)$$

where S_{ji} is a binary variable that equals 1 if V_j is located to run on PM; otherwise, it is 0.

3.3 Output Parameters

Solution should indicate the placement of each virtual machine V_j on the necessary physical machine S_i , considering the multi-objective optimization criteria. A placement is represented as a matrix $S = S_{ji}$ of dimension $(m \times n)$, where $S_{ji} \in \{0, 1\}$ indicates if the virtual machine V_j is located ($S_{ji} = 1$) or not ($S_{ji} = 0$) for execution on a physical machine S_i (i.e., $S_{ji}: V_j \rightarrow S_i$).

4 Solving MODM Problem

4.1 Scalarization Method

The general multi objective decision making problem can be presented as follows:

$$\begin{cases} g(x) = [g_1(x), g_2(x), g_3(x), g_4(x), g_5(x)] \\ \text{Subjectto } h_i(x) \text{ with } i \in \{1, 2, 3, 4, 5\} \end{cases} \quad (17)$$

The idea of finding a solution for (18) would be challenging since a single point that minimizes all objective functions simultaneously often doesn't exist. In this context, we normalize each objective function by computing the normalized objective function cost $\hat{g}_i(x)$ as:

$$\hat{g}_i(x) = \frac{g_i(x) - g_i(x)_{min}}{g_i(x)_{max} - g_i(x)_{min}} \quad (18)$$

Based on previous works [12], different scalarization methods can be used to solve the multi-objective optimization approaches as Benson's method, Weighted Tchebyshev method, Pascoletti-Serafini method, Weighted sum method (WS), Euclidean Distance (ED), etc. In [5], the efficient method for VMP placement was ED, while in [12], authors recommend the Tchebychev method, and as per [19] and [16], the often-used method is Weighted sum. In this paper, we consider the Conic Scalarization (CS) method detailed in [14]. The concept of this method is straightforward: (i) choosing preference parameters which consist of weights of objective functions and a reference point for these objectives and (ii) solving the scalar optimization problem.

Consider the multi-objective virtual machine placement that aim to simultaneously minimize the five objective functions as follows:

$$\text{Minimise } g(x) = [g_1(x), g_2(x), g_3(x), g_4(x), g_5(x)]$$

First, choose preference parameters:

- Weight vector $\omega = \omega_i$: denotes the importance degree associated to each i -th objective function for decision maker, where $i \in \{1, \dots, 5\}$.
- Reference point $r = (r_1, r_2, \dots, r_5)$: identified by decision maker to compute the minimal elements and can be chosen arbitrarily.

Second, choose an augmentation parameter α such that $(\omega, \alpha) \in \mathbb{C}$, where:

$$\mathbb{C} = ((\omega_1, \dots, \omega_5), \alpha) : 0 \leq \alpha \leq \omega_i, \omega_i > 0, i = \{1, \dots, 5\}$$

The scalar problem for the given parameters (ω, α) and r is:

$$\min \sum_{i=1}^5 \omega_i (\hat{g}_i(x) - r_i) + \alpha \sum_{i=1}^5 |\hat{g}_i(x) - r_i| \quad (19)$$

In this work, the weight ω_i is constant ($\frac{1}{5}$). In the case where $\alpha = 0$, the scalarization method (20) becomes that of the weighted sum method.

4.2 Algorithms

This subsection presents the proposed alternatives used in our experiments to solve the VMP problem comparing the four online algorithms (FF, FFD, BF, BFD) against the offline algorithm ACO based memetic algorithm with VM migration.

- **First-Fit (FF)**: This algorithm places VMs according to the first in first out (FIFO) basis where requested VMs are allocated on the first host with available resources [11].

- **First-Fit-Decreasing (FFD):** The FFD algorithm works in a similar way to the FF algorithm presented above. It aligns VMs in the decreasing order, then finds and places servers with available resources to place VMs [2].
- **Best-Fit (BF):** This algorithm assigns the VMs required on the first PM with the available capacity from a sorted list of PMs in ascending order by a rating associated to each PM [2].
- **Best-Fit-Decreasing (BFD):** This algorithm is similar to BF. The difference is only on selecting VM lists in decreasing order by inquired CPU resources [2].
- **Memetic Algorithm (MA):** The term Memetic Algorithm describes population based hybrid evolutionary algorithms that are coupled with local refinement strategies, more details are presented in [20].
- **Ant Colony Optimization (ACO):** The ACO based algorithm is introduced as an instance of the multi-dimensional bin-packing problem [17].

5 Test Environment

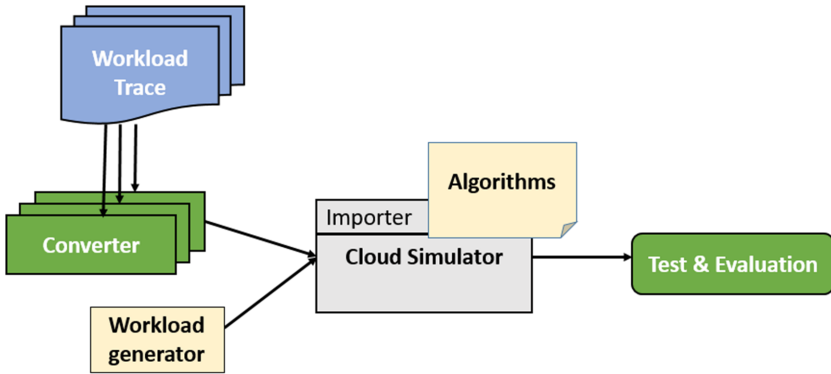


Fig. 3. Test environment

Experiments were conducted on a GNU Linux System with an Intel(R) Xeon(R) E3-1505M at 2.80 GHz CPU and 32 GB of RAM. Figure 3 shows an overview of the test environment. The proposed algorithms were developed by Java programming language. Unfortunately, it's not feasible (at least expensive) to test the VM placement based on real traces for a large-scale environment. In this paper, we use uncertain workload traces similar to real-world ones. We import data files of a particular format, to convert workload traces to the given format, and create test data in the given format. Physical resources comprise a diversified IaaS cloud, taking into account four categories of physical machines (i.e., small, medium, large, and xlarge), as presented in Table 1.

In this paper, we compare different algorithms in online and offline environments for various CPU load types. Simulation results presented in Table 2 prove the performance of BFD in both medium and high CPU load compared to other dynamic placement algorithms, while BF is the best one for low CPU load. On the other hand, one can see the ACO algorithm yields good results for all CPU load sizes. We can conclude that ACO based memetic algorithm is better than deterministic algorithms, otherwise the offline placement outperforms the online one. In Fig. 4, we compare the results of the conic scalarization method for MODM problem, considering different values of the augmentation parameter α (0.01, 0.05, 0.1, 0.15, 0.2) less than $\omega = \frac{1}{5}$. We see that the more α tends towards ω , the better results are obtained for all algorithms.

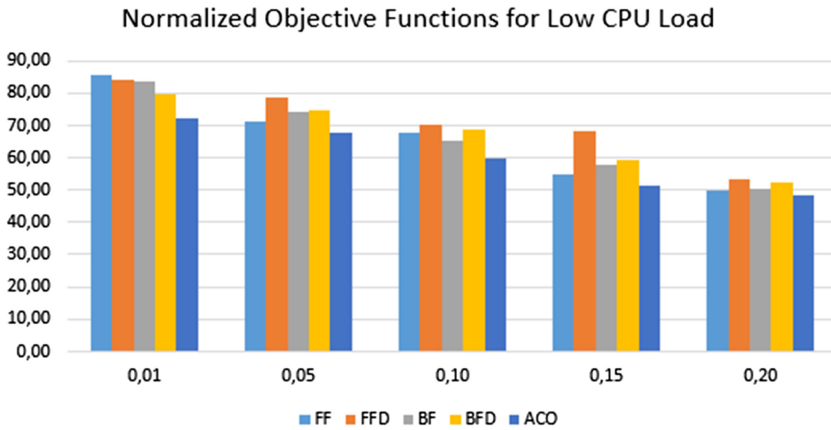


Fig. 4. Normalized objective functions for different values of augmentation parameter α (scale x100)

As a perspective, in the future work, we propose to combine the online and offline placement, and compare the CS method to other scalarization methods.

Table 1. Types of physical machines

PM characteristics	CPU	RAM	Network	$P_{max}(W)$
S.small	32	128	1000	500
S.medium	64	256	1000	1000
S.large	256	512	1000	2000
S.xlarge	512	1024	10000	4000

Table 2. Normalized objective functions of evaluated algorithms

Algorithm	Low CPU load	Medium CPU load	High CPU load
First-Fit	0.675	0.733	0.972
First-Fit decreasing	0.703	0.742	0.965
Best-Fit	0.653	0.723	0.846
Best-Fit decreasing	0.685	0.721	0.837
ACO	0.595	0.654	0.683

6 Conclusion

We consider the VMP problem as a multi-objective decision-making approach that aims to optimize simultaneously five objective functions: energy consumption, cost optimization, network traffic, resource utilization, and QoS. The optimization problem is solved based on a conic scalarization method. Simulation results show that ACO based-memetic gives good results compared to FF, BF, FFD, and BFD. However, offline algorithms can't be used in a pure dynamic environment. Therefore, our future will consider a combination of both offline and online placement algorithms.

References

1. Akintoye, S.B., Bagula, A.: Improving quality-of-service in cloud/fog computing through efficient resource allocation. *Sensors* **19**(6), 1267 (2019)
2. Atchukatla, M.S.: Algorithms for efficient VM placement in data centers: cloud based design and performance analysis. Master's thesis, Department of Computer Science and Engineering (2018)
3. Attaoui, W., Sabir, E.: Multi-criteria virtual machine placement in cloud computing environments: a literature review. [arXiv:1802.05113](https://arxiv.org/abs/1802.05113) (2018)
4. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Gener. Comput. Syst.* **28**(5), 755–768 (2012). Special Section: Energy Efficiency in Large-Scale Distributed Systems
5. Chamas, N., López-Pires, F., Baran, B.: Two-phase virtual machine placement algorithms for cloud computing: an experimental evaluation under uncertainty. In: 2017 XLIII Latin American Computer Conference (CLEI), pp. 1–10, September 2017
6. Chen, T., Zhu, Y., Gao, X., Kong, L., Chen, G., Wang, Y.: Improving resource utilization via virtual machine placement in data center networks. *Mob. Netw. Appl.* **23**(2), 227–238 (2018)
7. Choi, J.Y.: Virtual machine placement algorithm for energy saving and reliability of servers in cloud data centers. *J. Netw. Syst. Manage.* **27**(1), 149–165 (2019)
8. Cohen, R., Lewin-Eytan, L., Seffi Naor, J., Raz, D.: Almost optimal virtual machine placement for traffic intense data centers. In: 2013 Proceedings IEEE INFOCOM, pp. 355–359, April 2013

9. Dias, D.S., Costa, L.H.M.K.: Online traffic-aware virtual machine placement in data center networks. In: 2012 Global Information Infrastructure and Networking Symposium (GIIS), pp. 1–8, December 2012
10. Dong, J., Wang, H., Jin, X., Li, Y., Zhang, P., Cheng, S.: Virtual machine placement for improving energy efficiency and network performance in IaaS cloud. In: 2013 IEEE 33rd International Conference on Distributed Computing Systems Workshops, pp. 238–243, July 2013
11. Fang, S., Kanagavelu, R., Lee, B., Foh, C.H., Aung, K.M.M.: Power-efficient virtual machine placement and migration in data centers. In: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, pp. 1408–1413, August 2013
12. Gopu, A., Venkataraman, N.: Optimal VM placement in distributed cloud environment using MOEA/D. *Soft. Comput.* **23**(21), 11277–11296 (2019)
13. Gupta, M.K., Jain, A., Amgoth, T.: Power and resource-aware virtual machine placement for IaaS cloud. *Sustain. Comput. Inform. Syst.* **19**, 52–60 (2018)
14. Kasimbeyli, R., Ozturk, Z.K., Kasimbeyli, N., Yalcin, G.D., Icmen, B.: Conic scalarization method in multiobjective optimization and relations with other scalarization methods. In: Le Thi, H.A., Pham Dinh, T., Nguyen, N.T. (eds.) *Modelling, Computation and Optimization in Information Systems and Management Sciences*. AISC, vol. 359, pp. 319–329. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-18161-5_27
15. Kumar, D., Mandal, S.K.: Multi-objective virtual machine placement using improved teaching learning based optimization in cloud data centers. *Int. J. Appl. Eng. Res.* **12**, 10809–10815 (2017)
16. López-Pires, F., Barán, B.: Many-objective virtual machine placement. *J. Grid Comput.* **15**(2), 161–176 (2017)
17. Malekloo, M., Kara, N.: Multi-objective aco virtual machine placement in cloud computing environments. In: 2014 IEEE GLOBECOM Workshops (GC Workshops), pp. 112–116, December 2014
18. Meng, X., Pappas, V., Zhang, L.: Improving the scalability of data center networks with traffic-aware virtual machine placement. In: 2010 Proceedings IEEE INFOCOM, pp. 1–9, March 2010
19. Mishra, S., Sangaiah, A.K., Sahoo, M.N., Bakshi, S.: Pareto-optimal cost optimization for large scale cloud systems using joint allocation of resources. *J. Ambient Intell. Hum. Comput.* (2019). <https://doi.org/10.1007/s12652-019-01601-x>
20. Urselmann, M.: Derivative-free chemical process synthesis by memetic algorithms coupled to aspen plus process models. In: Kravanja, Z., Bogataj, M. (eds.) *26th European Symposium on Computer Aided Process Engineering*. Computer Aided Chemical Engineering, vol. 38, pp. 187–192. Elsevier (2016)
21. Zhang, J., He, Z., Huang, H., Wang, X., Gu, C., Zhang, L.: SLA aware cost efficient virtual machines placement in cloud computing. In: 2014 IEEE 33rd International Performance Computing and Communications Conference (IPCCC), pp. 1–8, December 2014
22. Zhao, L., Lu, L., Jin, Z., Yu, C.: Online virtual machine placement for increasing cloud provider's revenue. *IEEE Trans. Serv. Comput.* **10**(2), 273–285 (2017)