Multi-objective optimization for VM placement in homogeneous and heterogeneous cloud service provider data centers

Rym Regaieg, Mohamed Koubàa, Zacharie Alès, Taoufik Aguili

Multi-Objective Optimization for VM Placement in Homogeneous and Heterogeneous Cloud Service Provider Data Centers

Rym REGAIEG[†], Mohamed KOUBÀA[†], Zacharie ALES[‡], Taoufik AGUILI[†]

[†]: Laboratoire des Systèmes de Communications, École Nationale d'Ingénieurs de Tunis, Université de Tunis El Manar Tunis, Tunisia
{rym.regaieg, mohamed.koubaa, taoufik.aguili}@enit.utm.tn

[‡]: Unité de Mathématiques Appliquées, École Nationale Supérieure de Techniques Avancées, Université de Paris Saclay Palaiseau, France
zacharie.ales@ensta-paris.fr

Abstract

We address the Virtual Machine Placement (VMP) problem that arises in Cloud Service Providers data centers. We purpose, a Multi-Objective Integer Linear Programming model which aims at optimizing simultaneously the number of hosted Virtual Machines (VM), the resource wastage and the number of active Physical Machines (PM) in order to minimize power consumption. This new combination of objectives enables to maximize the client satisfaction rate with minimizing the Data Center (DC) operational costs. We modelize this problem with a multi-objective integer linear program and solve it through two different methods. The first method computes a unique solution for a given preference order over the objectives whereas the second computes a set of non-dominated solutions. Both methods are compared through extensive simulation scenarios. We consider two DC architectures: homogeneous DCs (i.e., a DC with PMs having the same amount of resources) and heterogeneous DCs. We study the impact of each DC configuration on the performances of the solutions. We show that the second method leads to solutions with a reduction of up to 30% over the number of used PMs and that the heterogeneous DCs outperforms the homogeneous one across all objectives.

Keywords: Virtual Machine Placement, MILP model, weighted sum method, Knee point

1 Introduction

Cloud Computing offers new models where configurable computing resources such as computing power, Internet applications, network and storage can be shared as services through the Internet. The computing resources are pooled to meet the demand of the users in Data Centers (DC) which are overseen by Cloud Service Providers (CSP). The CSP offers three main types of service models to its users known as Infrastructure as a Service (IaaS), Software as a Service and Platform as a Service [1, 2, 3]. In the IaaS model, virtualized computing resources (e.g., processing, memory, storage, network bandwidth, . . .) are offered as a service and their utilization is expected to comply with a Service Level Objectives (SLO) which entails the CSP to provide several group metrics such as security metrics [4] and quality of service metrics (performances, availability, reliability [5], . . .). The virtualization technology is the cornerstone of the IaaS model. This technology allows a physical computing system to be divided into separate and secure environments known as Virtual Machines (VMs where each VM can perform computing tasks [6]). These VMs are characterized by resource requirements (e.g., processing, memory, storage, . . .) which are defined either by the cloud user or by the CSP [7]. The process of selecting where the VMs should be placed in each Data Center Physical Machines (PM) is known as the Virtual Machine Placement (VMP) problem [8]. In light of the adaptable and versatile administration that the IaaS model provides a considerable number of companies, which, beforehand deployed their businesses locally, has now moved to the cloud.

The VMP process differs from one DC to another according to the CSPs' end-goals. For instance, some CSPs aim at minimizing the energy consumption in the DC as it represents one of the main causes for the high operational cost (OPEX). Other try to minimize the SLO violation [9] (i.e., resource capacities, response time, etc. . .) in order to avoid incurring penalties. Moreover, CSPs can also minimize the inter-server bandwidth consumption required for inter-VM communications due to the relatively scarce higher level bandwidth. With the expansion of cloud market, CSPs may have to combine two or more placement goals in order to stay competitive. Thus, in our previous work [10], we proposed a two-objective VMP solution, aiming at simultaneously maximizing the number of hosted VMs while minimizing the number of used PMs. Maximizing the number of hosted VMs may allow the CSP to obtain a higher customers' satisfaction rate. The minimization of the number of used PMs may lead to a lower power consumption. From our previous simulation results, we observed that even the proposed solutions which minimized the number of used PMs, had unused resources on their active PMs. Thus, the minimization of the number of used PMs does not necessarily always lead to the minimization of the resource wastage. This observation motivates us to propose a new multi-objective VMP problem where one of the objectives is dedicated to the minimization of resource wastage.

We modelize this problem with a new Multi-Objective Integer Linear Programming (MOILP) model which simultaneously optimizes the number of hosted VMs (O1), the amount of resource wastage (O2) and the number of used/active PMs (O3). Objective O1 also corresponds to the client's satisfaction rate. This is the first time, to our knowledge, that this combination of objectives is considered. We solve the MOILP model with two optimization methods. The first one, called Method 1, is the lexicographical preference/ordering method. It computes a VMP solution by successively solving the VMP problem for each objective according to the preference order O1, O2 and O3. For this purpose, constraints are added to ensure that the values of the previously considered objectives are not deteriorated. Objective O1 has the priority as the main concern of the CSP is to satisfy a maximal number of VMs requests. As the minimization of the number of used PMs generates huge amounts of wasted resources, objective O2 has a higher priority order than O3. We observe through extensive experiments that Method 1 provides poor compromise between the three conflicting objectives. As a consequence, we also consider the Weighted Sum (WS) method (Method 2). It computes a set of non-dominated solutions where each one is obtained through a weight combination of the objectives. The weights reflect the preferences towards the objectives. Providing a set of solutions allows a CSP to choose the one which suits the best its requirements. Whenever, a unique solution is required, we consider a knee point method to select a solution which provides a good trade-off between the objectives [12]. For each optimization method, we consider two DC architectures. A DC is said to *homogeneous* if all its PMs are identical, otherwise it is *heterogeneous*. Homogeneous DCs avoid additional maintenance costs due to the usage of different PMs types while, heterogeneous DCs enable better pricing and resource management modes. These two architectures allow to study the impact of the resource diversity on the VMP performances.

The rest of the paper is organized as follows. Section 2 formally presents the problem tackled in this paper. Related works are described in Section 3. In Section 4, we define the proposed model. Section 5 shows the simulation results and Section 6 concludes the paper.

2 Description of the problem

Let consider a set of VMs and a set of PMs where a machine m (physical or virtual) is defined by a vector $V_m = (C, R, S)$ which components represents its CPU, memory and storage, respectively. A set of VMs $V = \{(C_i, R_i, S_i)\}_{i=1}^{N}$ can be placed on a physical machine $V_m = (C_m, R_m, S_m)$ if the cumulated resources of the VMs do not exceed any of the PM resources (i.e., if $\sum_{i=1}^{N} C_i \leq C_m, \sum_{i=1}^{N} R_i \leq R_m$ and $\sum_{i=1}^{N} S_i \leq S_m$). A virtual machine placement problem consists in finding a valid placement of VMs on the PMs which optimizes a given placement goal. We assume that all the PMs are initially not hosting any VM. As the VM requests are known a priori, the problem is also called the offline VMP problem as opposed to the online/dynamic VMP problem. A VM request can be rejected (i.e., assigned to no PM).

Figure 1 shows an instance of a VMP problem with 3 VMs and 3 PMs (Figure 1(a)) having the same resource characteristics and one possible solution using only two PMs (Figure 1(b)), where the placement goal is to maximize the number of hosted VMs. The height of the white bar in Figure 1 (b) corresponds to the total amount of CPU initially available at the PM whereas the height of the black bar matches the amount of consumed resources. VM_1 and VM_2 are hosted by PM_1 whilst VM_3 is hosted by PM_2 due to a lack of memory and storage of PM_1. This VMP solution produces a large amount of wasted resources due to the non utilization of about 40% of the CPU resource of PM_1 and more than 50% of all the resources of PM_2. One of the contribution of this paper is to introduce a new objective dedicated to the minimization of resource wastage.

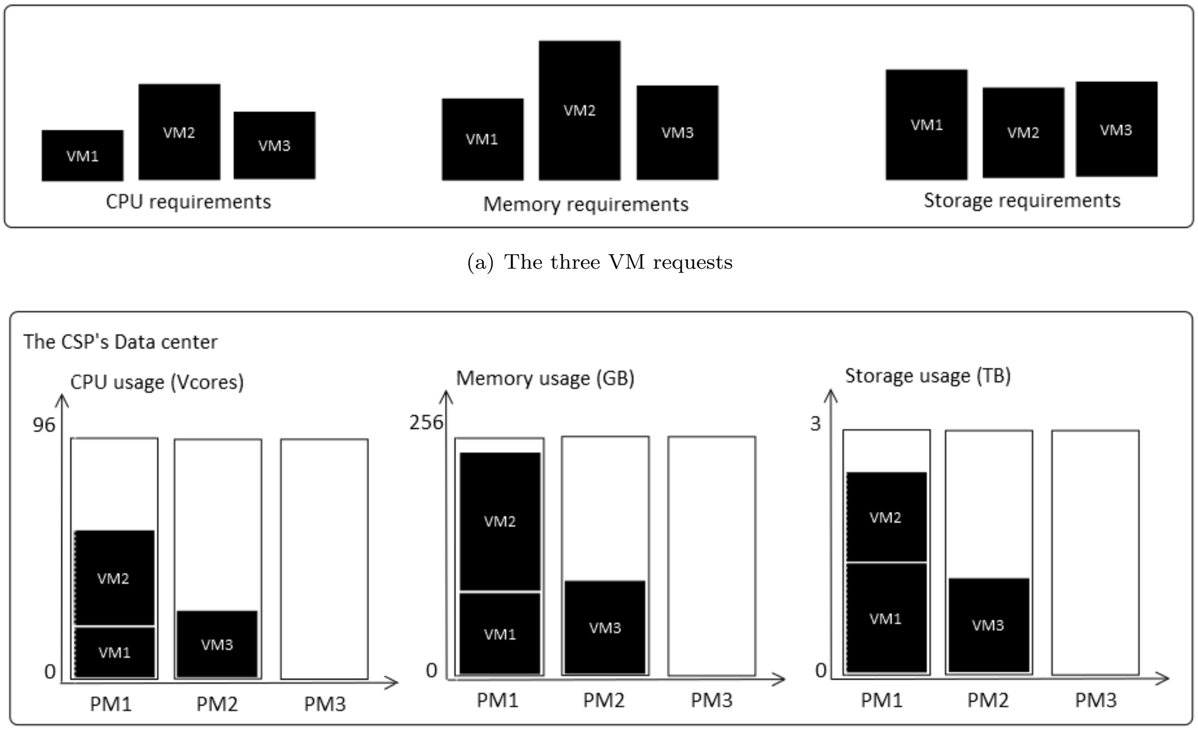
3 Related Work

3.1 VMP approaches

Different approaches have been considered to tackle VMP problems in terms of placement objectives and optimization methods.

3.1.1 Placement objectives

Multiple VMP metrics have been considered [8, 13]:



(a) The three VM requests

(b) An instance of VMP over the three PMs of the DC

Figure 1 The VMP problem.

- Energy consumption: CSPs try to minimize energy consumption, mainly by using as few PMs as possible. They can also favor the PMs with the best power efficiency since the energy consumption per instruction can be different. However, a PM's power consumption is not constant but depends on its load [14], which complicates the problem.

- Application performance: aggressive VM consolidation may lead to congestion or overload of a PM's resources. In that case, the accommodated VMs cannot obtain the amount of resources they would need, resulting in performance degradation of the applications running in those VMs, which in turn likely leads to violation of SLOs and thus financial penalty for the CSP. In addition, SLO violation is likely to adversely impacts customer satisfaction [20, 21].

- Economical revenue: the maximization of the CSP's economical revenue can be achieved by either minimizing the total economical penalties for SLO violations, minimizing the operational costs or even by maximizing the total profit for leasing resources [22, 30].

- Network traffic: the huge amount of intra-data center traffic is primarily generated by the VMs that are correlated to each other (i.e., VMs which are assigned to several PMs). Thus, the resource of bandwidth can become a bottleneck in the DC and cause multiple problems such as congestion, link overloads and connection disruptions [15]. The principal ways to minimize the network traffic are the minimization of the data transfer time, the minimization of the average traffic latency and the maximization of the network performance.

- Resource utilization: the resource utilization of servers (such as CPU, memory, disk, etc ...) is a critical performance metric in DCs because a poor resource allocation scheme could lead to higher operational costs. Typically, underutilized servers consume approximately 70% of their peak power, resulting in a higher power consumption in the DC [16]. Therefore, a resource utilization efficiency could be achieved by the minimization of the total amount of resource wastage or the maximization of the resource usage in the DC.

3.1.2 Optimization methods

Optimization methods fall under one of the following categories:

- Exact methods : they provide an optimal solution for the combinatorial optimization problem. However, they require a large execution time on large instances [15, 8].

- Heuristic methods : these methods could not guarantee the solutions optimality, but they may find a near-optimal solution within an acceptable execution time [15, 8].

3.2 The mono-objective VMP problem

The Mono-objective VMP problem has been well studied in the literature and aims at optimizing one objective function. Table 1 gives an overview on the most interesting solutions proposed in this context. In [19], Tang. et al., have considered minimizing power consumption. They propose an hybrid genetic algorithm to solve the energy-efficient VMP problem. The proposed solution is evaluated with VMs of random resource configurations in a heterogeneous DC. The authors have demonstrated that the proposed VMP approach was practical for the offline VM placement in both small and/or medium data centers. The results showed that the proposed algorithm works better than others algorithms of the literature and guarantees the VM performance. In [21], the authors have proposed a performance-aware VM placement algorithm to solve the VMP problem. The performances were evaluated with predefined VM configurations in a heterogeneous DC. In [22], the authors have considered maximizing the cloud provider revenues. An Integer Linear Programming formulation is proposed to compute the exact solution. The performances were evaluated with random VM configurations in a homogeneous DC. The authors have demonstrated that the proposed VMP approach was practical for the offline VM placement in both small and/or medium data centers.

3.3 The multi-objective VMP problem

The Multi-objective VMP problem (MOVMP) has been well studied in the literature and aims at optimizing multiple objective functions at a time. Table 1 gives a condensed overview of the principal MOVMP approaches proposed. We observe that:

- The combination of the maximization of resource utilization and the minimization of energy consumption is the predominant approach [24, 25, 26, 27, 28]. This can be explained by the fact that these two objectives help reducing the DC operational costs [17].
- Most of the works use heuristics and meta-heuristics such as Ant Colony Optimization (ACO) [33], Particle Swarm Optimization (PSO) [31] and Genetic Algorithms (GA) [32]. This is due to the NP-hardness of the VMP problem [8, 18, 28].
- Homogeneous DCs and predefined VM configurations are generally considered. This is due to the fact that the MOVMP problem is easier to model and solve with these types of DC and VM.

3.4 Contributions

The main contributions of this paper are given as follows:

- A Multi-Objective ILP model (MOILP) based on a new combination of objective functions is proposed to solve the VMP problem. The objectives are the maximization of the number of hosted VMs, the minimization of the amount of resource wastage and the minimization of the number of used PMs. This combination of objectives is chosen in order to provide to the CSPs a VMP solution achieving a higher client satisfaction rate with lower DC operational costs.
- Two optimization methods are used to solve the MOILP model. Method 1 computes an optimal VMP solution considering a preference order over the objectives. Its drawback is that it generally does not provide a good compromise between the conflicting objectives. Method 2 which computes a set of non-dominated solutions. Through a method called Knee point, a VMP solution which achieves a maximum trade-off between the considered objectives can be selected among the returned solutions.
- A comparative study is established between the two optimization methods.
- Both homogeneous and heterogenous DCs are considered to study the impact of the PM diversity on the performances.

Table 1 VMP approaches in the literature.

	Objective group	Optimization methods		Simulation context			
				DC Architectures		VM Configurations	
		Exact	Heuristic	Homogeneous	Heterogeneous	Predefined	Random
Mono-objective	Energy consumption minimization	[18]	[18, 19]	[18, 19]		[18, 19]	
	Performance maximization		[20, 21]	[20]	[21]	[21]	[20]
	Economical revenue maximization	[22, 23]		[22, 23]			[22, 23]
Multi-objectives	Energy consumption minimization	[24, 27, 28]	[24, 25, 26, 28, 29, 30]	[24, 28, 30]	[25, 26, 27, 29]	[25, 26, 27, 28, 29, 30]	[24]
	Network traffic minimization	[28]	[28, 31, 32]	[28, 31]	[32]	[28, 31]	[32]
	Economical revenue maximization	[27]	[30]	[30]	[27]	[27, 30]	
	Performance maximization	[24, 27, 28, 33]	[24, 25, 26, 28]	[24, 28]	[25, 26, 27] [33]	[25, 26, 27, 28, 33]	[24]
	Resource utilization maximization		[29, 31, 32, 33]	[31, 29]	[32, 33]	[31, 29, 33]	[32]

4 The Multi-Objective Integer Linear Programming Model

We propose to solve the MOILP model with two optimization methods. The first one, called Method 1, is the lexicographical preference/ordering method. Method 1 computes the VMP by successively solving the VMP problem for each objective according to the preference order O1, O2, O3. Constraints are added to ensure that the values of the previously considered objectives are not deteriorated. The second one is called the Weighted Sum (WS) method (Method 2) which computes a set of non-dominated VMP solutions where each one is obtained through the combination of objectives with a given set of weights. The main difference between the two methods is that Method 2 provides a set of non-dominated solutions which may lead to a higher trade-off between the considered objectives.

4.1 Notations

In the following, we use the following notations and typographical conventions:

- i and j as subscripts denote a virtual machine request and a physical machine index respectively.
- N denotes the number of VM requests. The i^{th} VM is denoted by v_i, and defined by the triplet (c_i, r_i, s_i) where c_i, r_i and s_i are the CPU, the memory and the storage requirements of the VM, respectively.
- M denotes the number of physical machines in the DC. The j^{th} PM, denoted by P_j is characterized by the triplet (C_j, R_j, S_j) where C_j, R_j and S_j are the CPU, the memory and the storage capacities of the PM, respectively.
- binary variable λ_{ij} is equal to 1, if v_i is hosted by P_j and 0, otherwise.
- binary variable ϕ_j is equal to 1, if there is at least one virtual machine hosted by P_j and 0, otherwise.

4.2 Lexicographical preferences method

In the first method, the objectives are optimized successively in three different MILP, as shown in Figure 2. The formulations associated with Step 1, Step 2 and Step 3 are given in Table 2.

Step 1 optimizes the VM-PM mapping with the objective of maximizing ψ_{max}, the number of hosted VM requests. Equation (2) ensures that each VM request v_i is hosted by at most one physical machine

P_j. Equation (3) ensures that the total amount of CPU consumed by the VMs hosted on P_j is at most equal to C_j. Equations (4) and (5) are similar to (3) for the CPU resource is replaced memory and storage resources, respectively. There may be multiple solutions with an optimal number of hosted VM requests. Step 2 selects one which in addition, minimizes the resource wastage, δ_{min}. The amount of wasted resources is computed as the total amount of unused resources on active PMs [34]. Equation (8) ensures that the number of hosted VM requests is still optimal. Equations (9) and (10) define ϕ_j variables. Once again, many equivalent solutions may exist at the end of Step 2. The last step selects one of them which, in addition, minimizes the total number of used PMs in the DC, θ_{min}. Equation (13) ensures that the total amount of wasted resources in the DC is not deteriorated.

Objective O_3 may seem redundant with O_2 as they both minimize the sum of ϕ_j but it is not. Indeed, an optimal solution for O_2 may not be optimal for O_3. To highlight this, we consider a simple example in which, for the sake of simplicity, we assume that a machine (physical or virtual) is only characterized by its CPU resource. Let us consider the two VMs and three PMs represented in Tables 3 and 4. Figure 3 shows two solutions S_1 and S_2 which are optimal for O_2.

Solution S_2 is not optimal for O_3, since it uses two PMs whereas S_1 only uses one. Consequently, O_2 does not necessarily minimize the number of PM used.

4.3 Weighted-Sum method

The weighted-sum is a generic method to obtain non-dominated solutions of a multi-objective problem. The objectives are aggregated in a linear combination as examplified in Table 5. In this table, W_1, W_2, W_3 are the weight coefficients associated to the number of hosted VMs, the amount of resource wastage and the number of used PMs objectives, respectively. In this article, we consider the weighted-sum approach implemented in the Julia package MultiJuMP. This method normalizes the objectives in order to avoid scaling deficiencies [43]. Each objective o is replaced by:

$$z(o) = \frac{o - f^{\min}}{f^{\max} - f^{\min}} \tag{14}$$

where $f^{\max}$ and $f^{\min}$ represent the maximal and minimal value that could be set to o, respectively. The combinations of coefficient weights are then obtained by creating a uniform grid. Each point on

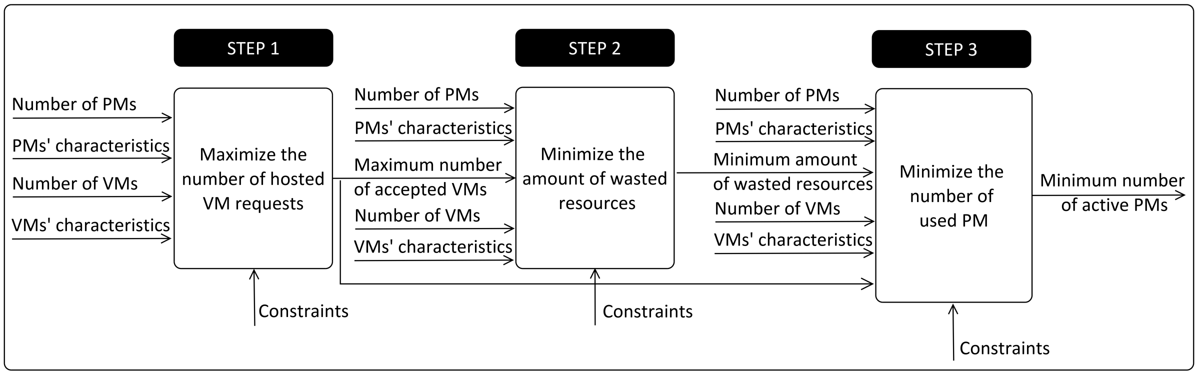


Figure 2 Method 1.

Table 2 The MOILP models of Method 1.

Step 1

Given N, M, C_j, R_j, S_j, c_i, r_i and s_i

Maximize $$\psi_{max} = \sum_{i=1}^{N} \sum_{j=1}^{M} \lambda_{ij} \tag{1}$$

Subject to :

$$\sum_{j=1}^{M} \lambda_{ij} \leq 1, \qquad \forall 1 \leq i \leq N \tag{2}$$

$$\sum_{i=1}^{N} c_i \lambda_{ij} \leq C_j, \qquad \forall 1 \leq j \leq M \tag{3}$$

$$\sum_{i=1}^{N} r_i \lambda_{ij} \leq R_j, \qquad \forall 1 \leq j \leq M \tag{4}$$

$$\sum_{i=1}^{N} s_i \lambda_{ij} \leq S_j, \qquad \forall 1 \leq j \leq M \tag{5}$$

$$\lambda_{ij} \in \{0, 1\}, \quad \forall 1 \leq i \leq N, \forall 1 \leq j \leq M \tag{6}$$

Step 2

Given N, M, D, C_j, R_j, S_j, c_i, r_i, s_i and ψ_{max}

Minimize

$$\delta_{min} = \sum_{j=1}^{M} \left(3\phi_j - \sum_{i=1}^{N} \left(\frac{c_i \lambda_{ij}}{C_j} + \frac{r_i \lambda_{ij}}{R_j} + \frac{s_i \lambda_{ij}}{S_j} \right) \right) \tag{7}$$

Subject to :

$$\psi_{max} \leq \sum_{i=1}^{N} \sum_{j=1}^{M} \lambda_{ij} \tag{8}$$

$$\lambda_{ij} \leq \phi_j, \qquad \forall 1 \leq i \leq N, \forall 1 \leq j \leq M \tag{9}$$

$$\phi_j \leq \sum_{i=1}^{N} \lambda_{ij}, \qquad \forall 1 \leq j \leq M \tag{10}$$

(2), (3), (4), (5) and (6)

$$\phi_j \in \{0, 1\}, \qquad \forall 1 \leq j \leq M \tag{11}$$

Step 3

Given N, M, D, C_j, R_j, S_j, c_i, r_i, s_i, ψ_{max} and δ_{min}

Minimize $$\theta_{min} = \sum_{j=1}^{M} \phi_j \tag{12}$$

Subject to :

$$\sum_{j=1}^{M} \left(3 - \left(\sum_{i=1}^{N} \left(\frac{c_i \lambda_{ij}}{C_j} + \frac{r_i \lambda_{ij}}{R_j} + \frac{s_i \lambda_{ij}}{S_j} \right) \right) \right) - 3 \left(M - \sum_{j=1}^{M} \phi_j \right) \leq \delta_{min} \tag{13}$$

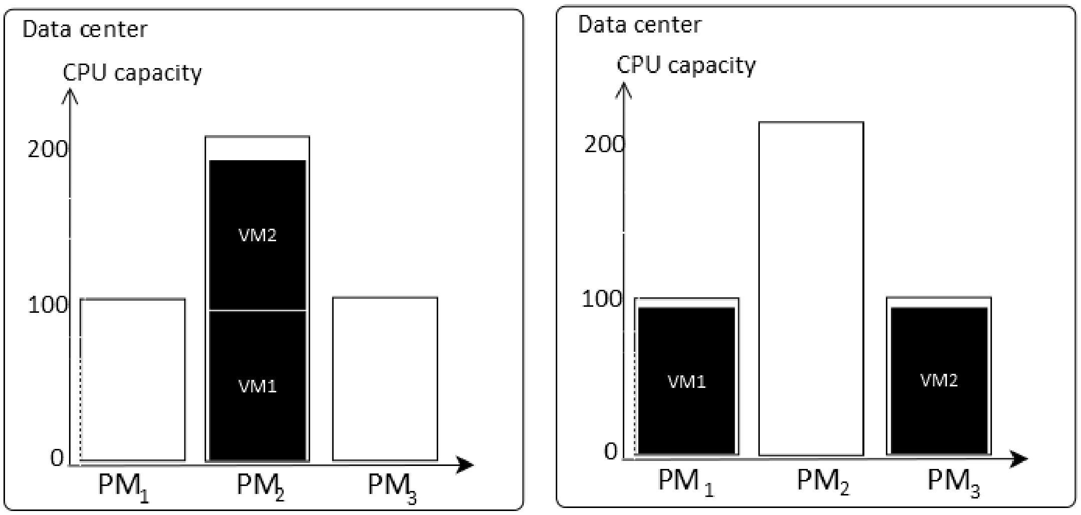
(2), (3), (4), (5), (6), (8), (9), (10) and (11)

Table 3 The VM configurations.

VM	CPU(units)
1	90
2	90

Table 4 The PM configurations

PM	CPU (units)
1	100
2	225
3	100



(a) A VMP solution S_1 with 20% of resource wastage and one used PM.

(b) A VMP solution S_2 with 20% of resource wastage and two used PMs.

Figure 3 The VMP problem

the grid where the sum of the coordinates components are equal to one, is selected as a combination of coefficients weights. Solving such a model for a given value of the weights leads to a non-dominated solution of the problem.

Weighted-sum method solves the aggregated problem for different values of the weights in order to generate a subset of the Pareto front. For each combination of weights generated, we solve the MOILP model given in Table 5 and obtain a VM-PM mapping which minimizes γ_{min}, the linear aggregation function of the objectives.

Table 5 The MOILP model of Method 2.

Given N, M, C_j, R_j, S_j, c_i, r_i , s_i, W_1, W_2 and W_3

Minimize $\gamma_{min} = -W_1\psi_{max} + W_2\delta_{min} + W_3\theta_{min}$ (15)

Subject to :

(2), (3), (4), (5), (6), (9), (10) and (11)

5 Simulation Results

5.1 Simulation parameters

We generate VMP instances with 20 to 220 VM requests. For each considered value of N, we randomly generate a set of VMs from a predefined set of VM types referred to as Small (S), Medium (M), Large (L) and XLarge (XL) in accordance with Amazon Elastic Computing Cloud (EC2). Thus, in the following, each value of N is associated with one instance. The characteristics of the VM types are given in Table 6 [35]. Figure 4 represents the number of generated VMs of each type in the instances. We consider the two DC configurations depicted in Table 7. The homogeneous DC contains 5 identical PMs while the heterogenous DC is composed of 8 PMs of 4 different types. The PM configurations of the homogeneous and heterogeneous DCs are given in Table 8. Note that both DC configurations have the same total amount of resources even if the total number of PMs are different.

The two optimization methods are implemented through Julia Language (JL) [40]. We use the MultiJuMP package which implements the weighted-sum method described in Section 4.3 [41]. Both methods use CPLEX 12.6.3 [42] to solve the MOILP models on a Linux server with 32 Intel Xeon Cores 2.0 GHz and 256GB of RAM.

Table 6 The VM configurations [35].

VM	CPU(Core)	RAM(GB)	DISK(GB)
S	2	2	20
M	2	4	40
L	2	8	80
XL	4	16	120

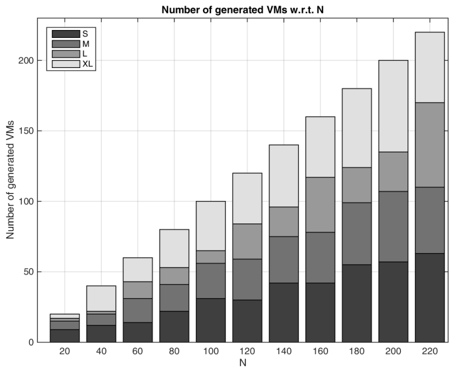


Figure 4 Number of generated S, M, L and XL w.r.t. N.

Table 7 The DC architectures.

DC architectures				
	Number of PM1	Number of PM2	Number of PM3	Number of PM4
Homogeneous DC	0	0	0	5
Heterogeneous DC	2	2	1	3

Table 8 The PM configurations [36, 37, 38, 39].

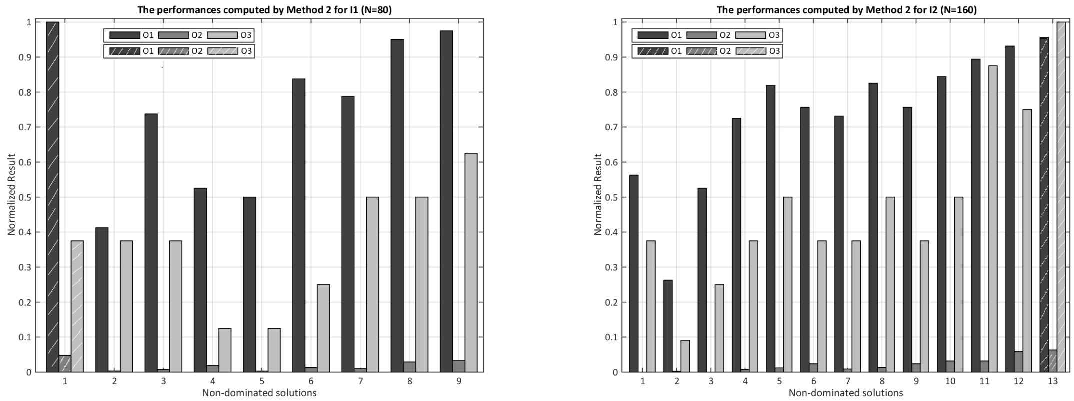
PM Type	CPU (Core)	RAM (GB)	DISK (GB)
DELL P. EDGE R440 (PM1)	12	32	320
HPE P. DL380 Gen9 (PM2)	42	112	1120
HPE P. DL580 Gen10 (PM3)	60	160	1600
DELL P. EDGE R510 (PM4)	84	224	2240

5.2 Discussion

In the following, all the objectives are normalized using Equation 14.

5.2.1 MOILP solutions

We remind that O1 corresponds to the maximization of the number of hosted VMs, O2 corresponds to the minimization of the amount of resource wastage and O3 corresponds to the minimization of the number of used PMs.



(a) Solutions of I_1.

(b) Solutions of I_2.

Figure 5 Objectives values of the non-dominated solutions obtained with Method 2 for two instances I_1 and I_2.

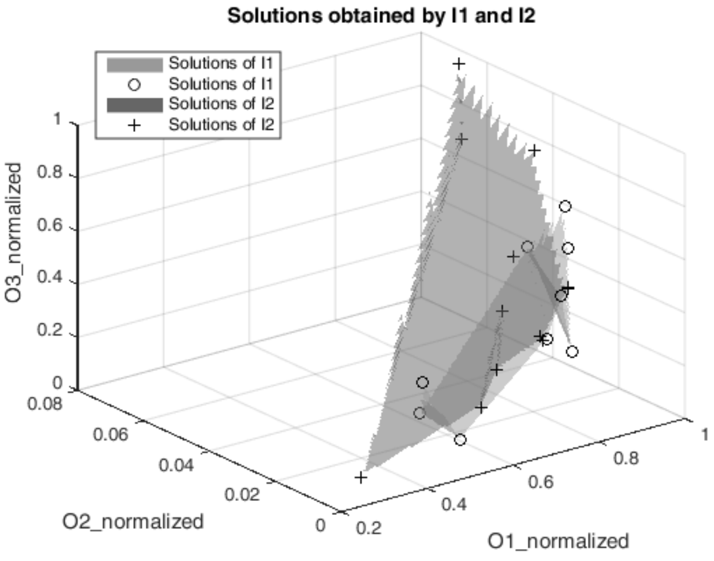


Figure 6 Non-dominated solutions obtained through I_1 and I_2.

Figure 5 represents the non-dominated solutions obtained with Method 2 on two heterogeneous instances I_1 and I_2 of size 80 and 160, respectively. The resources in the DC enable to satisfy all the VM requests in I_1, but not in I_2. Figure 5 corresponds to the value of the objectives of the 9 solutions obtained for I_1 and the 13 solutions of I_2. Each solution is represented by three bars which correspond from left to right to the normalized value of O1, O2 and O3, respectively. We can observe that at least 5% of the VMs are rejected in I_2. The number of solutions obtained in I_2 is higher than in I_1. This may be explained by the fact that the VM resource requirements exceed the resource capacities of the DC, leading to more possibilities for distributing VMs across the PMs. Figure 6 shows the solutions obtained for both instances. The solutions of I_1 covers a wider area than that of I_2. The hashed groups of three bars represent the VMP solution obtained by Method 1 for each instance. Method 2 provides a variety of trade-offs over the performances of the objectives. For example, for I_1, the selection of the 1^{st} solution leads to a preference towards the number of hosted VMs. A gain of up to 59% can be achieved for this objective. Solution 5 leads to a preference towards the amount of resource wastage which leads to an improvement of up to 25%. Finally, the selection of the fourth solution leads to a preference towards the number of used PMs. A gain of up to 80% can be obtained. Similar observations are obtained when considering a homogeneous DCs.

Figures 7, 8 and 9, respectively, show the number of variables, of constraints and the execution times required to model and solve the instances of each DC architecture. The NP-hardness of the problem explains why the execution time quickly increases with N.

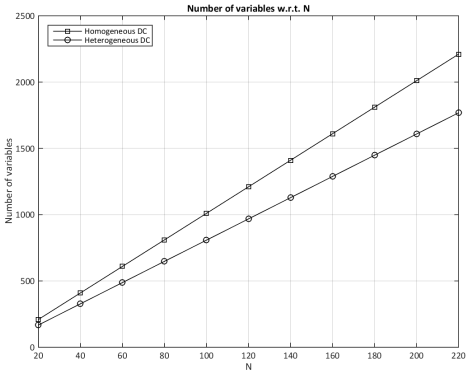


Figure 7 Number of variables w.r.t. N.

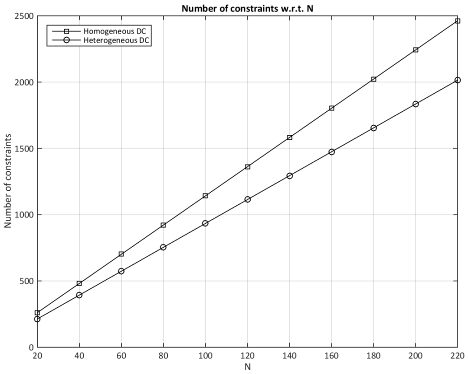


Figure 8 Number of constraints w.r.t. N.

5.2.2 Performance comparison between Method 1 and Method 2

In order to compare the performances of Method 1 and Method 2, we select one solution provided by Method 2. For this purpose, we choose the knee point which is likely to represent the maximal trade-off between the objectives [44]. Multiple methods for knee point recognition supporting high dimensional spaces, are proposed in the literature [45]. We use a trade-off worth metric defined in [44]. The expression of the trade-off information for a pair of optimal solutions x_i and x_j [45] is given by:

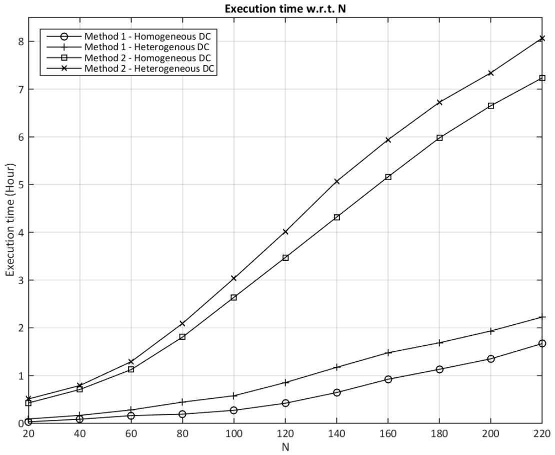


Figure 9 Execution time w.r.t. N.

$$T(x_i, x_j) = \frac{\sum_{m=1}^{M} \max\left[0, \frac{f_m(x_j) - f_m(x_i)}{f_m^{\max} - f_m^{\min}}\right]}{\sum_{m=1}^{M} \max\left[0, \frac{f_m(x_i) - f_m(x_j)}{f_m^{\max} - f_m^{\min}}\right]}$$

where $f_m(x_i)$ corresponds to the m^{th} objective value of solution x_i and $f_m^{\max}/f_m^{\min}$ the maximal/minimal value of the m^{th} objective in the set of non-domianted solutions S. In the above equation, all the values have been normalized. Based on this equation, the following expression is used to compute the worth of a solution x_i, in the S [44]:

$$\mu(x_i, S) = \min_{x_j \in S \setminus x_i \neq x_j, x_j \neq x_i} T(x_i, x_j)$$

where $\mu(x_i, S)$ expresses the least amount of improvement per unit of deterioration by substituting any alternative x_j from the S with x_i. The solution representing the knee point is $\underset{x_i \in \mathcal{S}}{\text{argmax}}\ \mu(x_i, S)$.

Figure 10 shows the performances of the MOILP solutions computed by Method 1 and Method 2 in the two DC architectures. Each group of six bars in this plot shows the performances of both optimization methods for each value of N, over one instance. The two first bars correspond to the normalized number of hosted VMs computed by Method 1 and Method 2, respectively. The next two bars depict the normalized amount of resource wastage and the last two bars present the normalized number of used PMs. Method 2 achieves average gains of 34% and 10% over Method 1 in the homogeneous DC over the amount of resource wastage and the number of used PMs, respectively, while an average loss of 3% over the number of hosted VMs is observed. For the heterogeneous DC, Method 2 achieves an average gain of 30% over the number of used PMs and average losses of 2% and 9% over the number of hosted VMs and amount of wasted resources, respectively. These last results, stress the effectiveness of Method 2 over Method 1 as it achieves, in both DC architectures, a significant improvement over the number of used PMs with only a small deterioration over the other objectives.

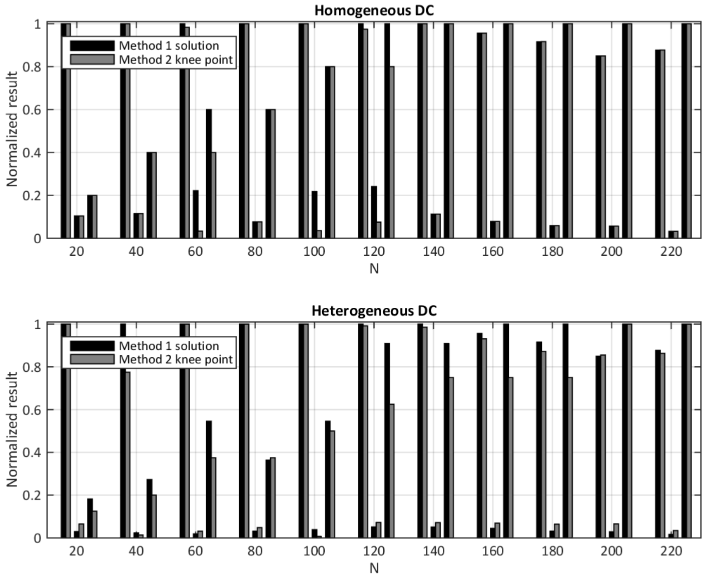


Figure 10 Performances of Method 1 and Method 2 for the two DC architectures.

5.2.3 The impact of DC architectures on the performances of MOILP model

As Method 2 showed better results than Method 1 in the previous section, we consider the solutions provided by Method 2 to compare the performances of the two DC architectures. Two DC configurations are considered for each architecture, which lead to a total of four DC configurations C1, C2, C3 and C4 represented in Table 9 and Table 10.

Table 9 The homogeneous DC configurations.

The homogeneous DC configurations.				
	Number of PM1	Number of PM2	Number of PM3	Number of PM4
C1	0	10	0	0
C2	0	0	0	5

Table 10 The heterogeneous DC configurations.

The heterogeneous DC configurations				
	Number of PM1	Number of PM2	Number of PM3	Number of PM4
C3	2	2	1	3
C4	4	4	2	1

Figure 11 shows the computed solutions for each DC configuration on an instance with 100 VMs. We can see that more solutions are obtained for the heterogeneous DCs. Indeed, the heteregeouns DCs are composed of PMs with different resource configurations which enable more variability in the placement of the VMs. Such a variability results in the generation of different trade-offs over the considered objectives and hence, a higher number of non-dominated VMP solutions are obtained.

Figure 12 plots the knee points obtained on four instances of size 80, 100, 120 and 140 for each of the four DC configurations. For each value of N, a quadruplet of bars presents the performances of an objective for the four DC configurations. From left to right the quadruplets correspond to the

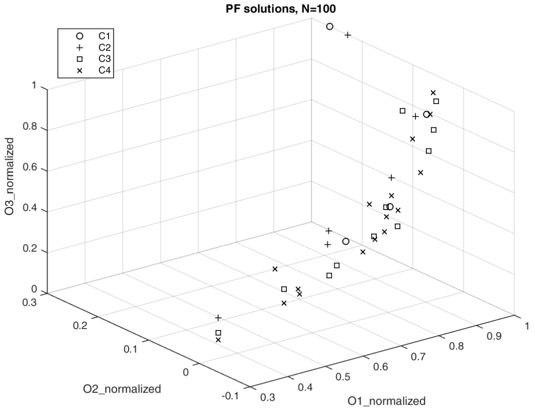


Figure 11 Non-dominated solutions computed for the four DC configurations, $N = 100$.

normalized number of hosted VMs, the normalized amount of resource wastage and the normalized number of used PMs. Heterogeneous DCs achieve average gains of up to 2%, 27% and 34% over O1, O2 and O3, respectively, over homogeneous DCs. This is mainly due to the diversity of the PM types in the heterogeneous DCs which enables to better meet the VMs requirements and leads to a lower amount of remaining resources on the PMs. To clarify this last result, we plot Figure 13 which represents the number of hosted VMs on each PM for the four DC configurations for the instance N=100. Each group of bars shows the results obtained for a DC configuration with respect to the number of available PMs numbered from 1 to M. We can observe from Figure 13 that, in the heterogeneous DCs, each PM type hosts a different combination of VMs: PM1 hosts a VM combination of S, M and L, PM3 hosts a VM combination of S, M, L and L and PM4 almost hosts a VM combination of S, M, XL and XL. This placement variability across the PM leads to better performances over the three objectives.

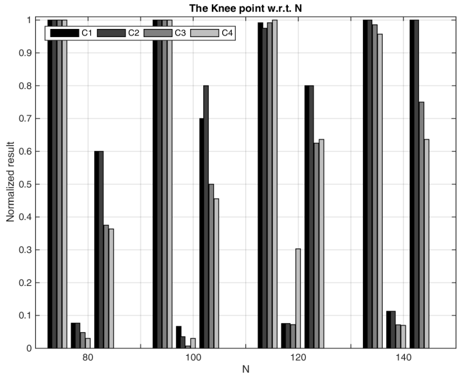


Figure 12 The knee points for the four DC configurations w.r.t. N.

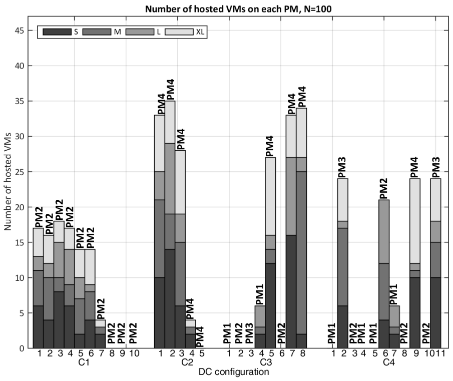
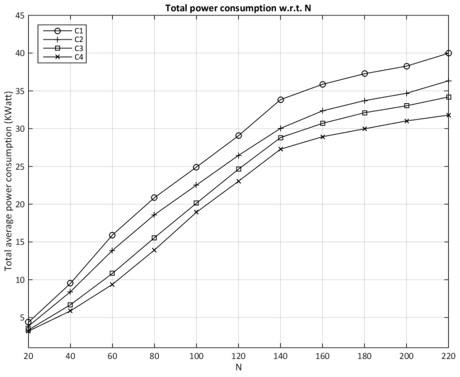


Figure 13 Number of hosted VMs on each PM, for the instance $N = 100$.

Figure 14 represents the total power consumption w.r.t. N. The total energy consumption of a DC configuration is the sum of all the energy consumption of its PMs in the DC. The power consumption of a given PM is computed according to [34]:

$$PC = P_{CPUIDLE} + (P_{CPUMAX} - P_{CPUIDLE})U$$

where $P_{CPUIDLE}$ and P_{CPUMAX} respectively represent the CPU idle and maximum processor power and U denotes the CPU utilization. For each PM configuration, the $P_{CPUIDLE}$ and P_{CPUMAX} are given in Table 11. From Figure 14, we can see that the total amounts of power consumption in the heterogeneous DCs are lower than the ones of the homogeneous DCs. An average gain of up to 23% is achieved. This is mainly due to the number of active PMs and the PM types comprising the DC. To highlight this, we plot Figure 15 which shows the CPU usage on each PM for the four DC configurations. The height of the white bar shows the total amount of CPU of a PM. The height of the gray bar shows the amount of CPU consumed by the hosted VMs. Figure 15 shows that, for the hosting of 100 VMs, C1 and C2 use seven and four PMs whose types are PM2 and PM4 respectively, whereas C3 and C4 use four and five PMs of types {PM1, PM4, PM4, PM4} and {PM3, PM2, PM1, PM4, PM3}, respectively. C1 uses PMs with a moderate power consumption level. However, the high number of used PMs leads to a high power consumption in the DC. For C2, the number of used PMs is comparatively low, however, the high power consumption level of each PM results in a higher power consumption in the DC. In the heterogeneous DC, both configurations use comparatively a medium number of PMs with various of power consumption levels (low, moderate and high).



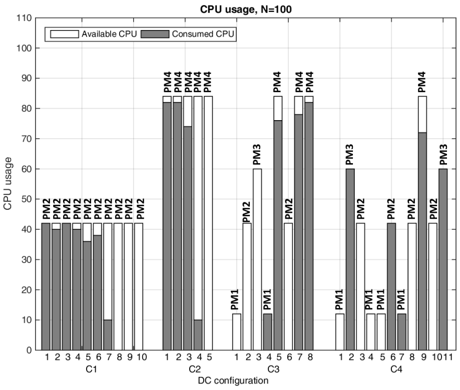


Figure 14 Total power consumption w.r.t. N. **Figure 15** CPU usage for each PM, N=100.

Table 11 The PM Power Consumption [16].

PM Type	$P_{CPU-IDLE(Watt)}$	$P_{CPU-MAX(Watt)}$
PM1	300	900
PM2	560	1100
PM3	800	1380
PM4	2100	2700

From the obtained results, we draw the following conclusions:

- Method 2 provides several non-dominated solutions to the CSP.
- Up to 30% less PMs are used with Method 2 for a loss of up to 2% and 9% over the number of hosted VMs and the amount of resource wastage, respectively. Such an outcome constitutes a suitable trade-off between the objectives.
- The solutions obtained by the MOILP model are more efficient in heterogeneous DCs compared to homogeneous DCs. The gains are of up to 2%, 27% and 34% over O1, O2 and O3, respectively.
- The heteregeouns DCs consume lower amounts of power than homogeneous DCs. An average gain of up to 23% is obtained.

6 Conclusion and Future Work

In this paper, we propose a new Multi-Objective ILP model to address the VMP problem in CSP DCs with homogeneous and heterogeneous PM types. The objectives are the maximization of the number of hosted VMs, the minimization of the amount of resource wastage and the minimization of the number of used PMs. We propose two optimization methods to solve the MOILP model. Through extensive simulations scenarios, we first show the effectiveness of Method 2 over Method 1 which leads to a better trade-off between the objectives. Finally, we observe that the heterogeneous DCs achieve better performances in terms of number of hosted VMs, amount of resource wastage, number of used PMs and amount of power consumption thanks to the diversity of the PM types. These results will help CSPs to reduce the DC operational costs while keeping a high client satisfaction rate. Due to the NP-hardness of the problem, exact approaches are quickly limited in the size of the VMP instances they can solve. Consequently, future work will focus on how to solve the VMP problem in real sized DCs using both heuristics and meta-heuristics.

References

[1] Buyya R, Yeo C.S, Venugopal S, Broberg J, Brandic I (2009) Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Future Generation Computer Systems 25(6):599-616

[2] Lombardi F, Di Pietro R (2011) Secure virtualization for cloud computing. Network and Computer Application 34(4):1113-11122

[3] Zhang Q, Cheng L, Boutaba R:Cloud computing (2010) state-of-the-art and research challenges. Internet Services and Applications 1(1):7-18

[4] Hoehl M (2020) Proposal for standard Cloud Computing Security SLAs - Key Metrics for Safeguarding Confidential Data in the Cloud. https://www.sans.org/reading-room/whitepapers/cloud/proposal-standard-cloud-computing-security-slas-key-metrics-safeguarding-confidential-data-cloud-35872. Accessed November 2020

[5] Serrano D, Bouchenak S, Kouki Y, Ledoux T, Lejeune J, Sopena J, Arantes L, Sens P (2013) Towards QoS-Oriented SLA Guarantees for Online Cloud Services. International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp 50-57.

[6] Tharam D, Chen W, Elizabeth CH (2014) Cloud computing: issues and challenges. International conference on advanced information networking and applications. IEEE, pp 27-33.

[7] Mell P, Grance T (2019) The NIST Definition of Cloud Computing. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/
nistspecialpublication800-145.pdf. Accessed November 2019

[8] Lopez-Pires F, Baran B (2015) Virtual Machine Placement Literature. Computing Research Repository (CoRR). https://arxiv.org/abs/1506.01509

[9] NIST Cloud Computing Reference Architecture and Taxonomy Working Group (2019) Cloud Computing Service Metrics Description. https://www.nist.gov/sites/default/files/documents/itl/
cloud/RATAX-CloudServiceMetricsDescription-DRAFT-20141111.pdf. Accessed November 2019

[10] Regaieg R, Koubàa M, Osei-Opokou E, Aguili T (2018) A Two Objective Linear Programming Model for VM Placement in Heterogeneous Data Centers. International Symposium on Ubiquitous Networking. Springer, pp 167-178.

[11] Cvetkovic D, Parmee I (1998) Evolutionary design and multi–objective optimisation. Proceedings of the 6th European Congress on Intelligent Techniques and Soft Computing (EUFIT) pp 397-401.

[12] Stanimirovic I, Zlatanovic M, Petkovic M (2011) On the linear weighted sum method for multi-objective optimization. Facta Acta Univ 26(4):49-63.

[13] Laghrissi A, Taleb T (2018) A survey on the placement of virtual resources and virtual network functions. In IEEE Communications Surveys and Tutorials 21(2):1409-1434.

[14] Rodero I, Viswanathan H, Lee E.K, Gamell M, Pompili D, Parashar M (2012) Energy-efficient thermal-aware autonomic management of virtualized HPC cloud infrastructure. Grid Computing 10(3):447–473.

[15] Attaoui W, Sabir E (2018) Multi-Criteria Virtual Machine Placement in Cloud Computing Environments. A literature Review. Review: arXiv preprint arXiv:1802.05113.

[16] Beloglazov A, Buyya R (2012) Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency and Computation: Practice and Experience 24(13):1397-1420.

[17] Yousafzai A, Gant A, Noor R.Md (2017) Cloud resource allocation schemes: review, taxonomy, and opportunities. Knowledge and Information Systems. Springer 50(2):347-381.

[18] Aydina N, Muterb I, Birbilc S-I (2019) Bin Packing Problem with Time Dimension: An Application in Cloud Computing. Preprint submitted to Elsevier.

[19] STang M, SPan S (2014) A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers. Neural Processing Letters pp 1–11.

[20] Lu K, Yahyapour R, Wieder P, Kotsokalis, C, Yaqub E, Jehangiri A.I (2013) Qos-aware vm placement in multi-domain service level agreements scenarios. In 6th International Conference on Cloud Computing (CLOUD). IEEE, pp 661–668.

[21] NAIR S.JA and NAIR T. R (2019) Performance degradation assessment and VM placement policy in cloud. Electrical and Computer Engineering 9(6):2088-8708.

[22] Shi L, Butler B, Wang R, Botvich D, Jennings B (2012) Optimal placement of virtual machines with different placement constraints in IAAS clouds. Symposium on ICT and Energy Efficiency and Workshop on Information Theory and Security pp 202-206.

[23] Addya S.K, Turuk A.K, Bibhudatta S (2017) Simulated annealing based VM placement strategy to maximize the profit for Cloud Service Providers. Engineering science and technology 20(4):1249-1259.

[24] Wenying Y, Chen Q (2014) Dynamic placement of virtual machines with both deterministic and stochastic demands for green cloud computing. Mathematical Problems in Engineering. Hindawi, pp 11.

[25] Mollamotalebi M, Hajireza S (2017) Multi-objective dynamic management of virtual machines in cloud environments. Cloud Computing 6(1): pp 16.

[26] Sultan A, Hamdaoui B (2018) Energy-Aware Resource Management Framework for Overbooked Cloud Data Centers with SLA Assurance. In Global Communications Conference (GLOBECOM). IEEE, pp 1-6.

[27] Guerout T, Gaoua Y, Artigues C, Da Costa G, Lopez P, Monteil T (2017) Mixed integer linear programming for quality of service optimization in Clouds. Future Generation Computer Systems. Elsevier, pp 1-17.

[28] Ihara D, Lopez-Pires F, Baran B (2015) Many-Objective Virtual Machine Placement for Dynamic Environments. In the 8th International Conference on Utility and Cloud Computing (UCC), pp 75-79.

[29] Gao Y, Guan H, Qi Z, Hou Y, Liu L (2013) A multi-objective ant colony system algorithm for virtual machine placement in cloud computing. Computer And System Sciences 70(8):1230-1242

[30] Shi L, Butler B, Wang R, Botvich D, Jennings B (2013) Provisioning of requests for virtual machine sets with placement constraints in IaaS clouds. International Symposium on Integrated Network Management. IEEE, pp 499-501.

[31] Luo J, Song W, Yin L (2018) Reliable Virtual Machine Placement Based on Multi-Objective Optimization with Traffic-Aware Algorithm in Industrial Cloud. IEEE Access 6:23043-23052.

[32] Patil J.T, Adamuthe A.C (2017) Solving Multi-objective Virtual Machine Placement in Cloud Computing Using NSGA-II. In National Conference for Engineering Post Graduate Students RIT NConPG-17, pp 182-187.

[33] Ma F, Liu F, Liu Z(2012) Multi-objective optimization for initial virtual machine placement in cloud data center. Information and Computational Science, 9(16):5029-5038

[34] Xu J, Fortes J.A.B (2010) Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In Conference on Cyber, Physical and Social Computing Green Computing and Communications. IEEE, pp 179-188.

[35] AWS (2020). https://aws.amazon.com/fr/ec2/instance-types/. Accessed November 2020

[36] HPE ProLiant DL380 Gen9 Server (2020). https://www.itcreations.com/hp/HP-ProLiant-DL380-Gen9-Server.asp. Accessed November 2020

[37] DELL PowerEDGE R440 Server (2020). https://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/poweredge-r440-spec-sheet.pdf. Accessed November 2020

[38] HPE ProLiant DL580 Gen10 Server (2020). https://www.hpe.com/us/en/product-catalog/servers/proliant-servers/pip.specifications.hpe-proliant-dl580-gen10-server.1010192779.html. Accessed November 2020

[39] DELL PowerEDGE R510 Server (2020). https://www.dell.com/support/home/fr/fr/frbsdt1/product-support/product/poweredge-r510/manuals. Accessed November 2020

[40] MultiJuMP (2020). https://julialang.org/. Accessed November 2020

[41] MultiJuMP (2020). https://github.com/anriseth/MultiJuMP.jl. Accessed November 2020

[42] IBM CPLEX Optimizer (2020). http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/. Accessed November 2020

[43] Bechikh S, Datta R, Gupta A (2016) Recent advances in evolutionary multi-objective optimization. Springer

[44] Rachmawati L, Srinivasan D (2009) Multiobjective evolutionary algorithm with controllable focus on the knees of the pareto front. IEEE Transactions on Evolutionary Computation 13(4):810-824.

[45] Bechikh S (2012) Incorporating decision maker's preference information in evolutionary multi-objective optimization. A PhD thesis University of Tunis.