Advances on Many-Objective Resource Allocation for Elastic Software-Defined Datacenters

Fabio Lopez-Pires
Itaipu Technological Park
fabio.lopez@pti.org.py
Paraguay

Abstract

In cloud computing resource allocation, Virtual Machine Placement (VMP) is one of the most studied problems with several possible formulations and different optimization criteria. The present work summarizes main advances focused on studying Many-Objective Virtual Machine Placement (MaVMP) problems. As first contributions, novel taxonomies were proposed for VMP problems in cloud computing environments, in order to gain a systematic understanding of the existing approaches. Additionally, first formulations of MaVMP problems were proposed in: (1) static MaVMP for initial placement, (2) semi-dynamic MaVMP with reconfiguration of VMs and (3) dynamic two-phase MaVMP for complex cloud computing environments under uncertainty. Considering the novelty of the proposed formulations, several methods and algorithms were also proposed to address main identified issues on solving each particular MaVMP problem. Experimental results prove the correctness, effectiveness and scalability of the proposed methods and algorithms in different experimental scenarios when comparing to state-of-the-art and industry alternatives. Open challenges for further advance of the area are discussed.

Keywords: Virtual Machine Placement, Many-Objective Optimization, Resource Allocation, Cloud Computing Datacenters, Elasticity, Overbooking.

1 Introduction

Significant research challenges for delivering computational resources as a fifth utility (like water, electricity, gas, and telephony) has already been identified [1]. In this context, cloud computing datacenters deliver infrastructure (IaaS), platform (PaaS) and software (SaaS) as services, available to tenants in a pay-as-you-go basis [2]. When cloud computing datacenters dynamically provide millions of *virtual machines* (VMs) to tenants in current cloud computing markets, achieving an efficient resource management for IaaS service model operations could be considered as one of the most relevant challenges, including important research topics such as: resource allocation, resource provisioning, resource mapping and resource adaptation [3]. Additionally, other research topics such as admission control and proactive elasticity could also represent relevant open challenges related to efficient resource management in cloud computing datacenters [4].

The research summarised in this paper focused on resource allocation, specifically in one of the most studied problems for resource allocation in cloud computing datacenters: the process of selecting which VMs should be hosted at each *physical machine* (PM) of a cloud computing infrastructure, known as *Virtual Machine Placement* (VMP). Several research articles in the specialized literature demonstrated that solving the VMP problem for efficient allocation of resources in cloud computing datacenters could significantly improve energy-efficiency, *quality of service* (QoS), carbon dioxide emissions, among other advantages; all of them with economical [5] and ecological impact [6].

Beloglazov and Buyya proposed in [7] four different sub-problems for resource allocation in cloud computing datacenters: (1) determining when a PM is overloaded, requiring migration of VMs from this PM; (2) determining when a PM is underloaded, requiring migration of all VMs from this PM and switching the PM to sleep mode; (3) selecting VMs to migrate from an overloaded PM; and (4) finding a new placement of the VMs selected for migration, considering overloaded and underloaded PMs. Additionally, a conceptual architecture considering the most studied problems related to resource allocation in cloud computing datacenters is presented by Elmroth et al. in [4], where placement of admitted services (composed by VMs) could be solved considering different criteria and requirements.

In cloud computing datacenters, there are several criteria that can be considered when selecting a solution for a VMP problem, depending on management policies and optimization objectives. These criteria can even change from one period of time to another, which implies a variety of possible environments, formulations and objectives to be considered for optimization of the VMP problem. As part of the summarised research, nearly 60 different objective functions were identified in the VMP literature [8, 9, 10]. Taking into account the large number of existing objective functions, providers of cloud computing datacenters must be able to formulate the VMP problem as a *Pure Multi-Objective Optimization Problem* (PMO), optimizing more than just one objective function at a time. It is worth remembering that PMOs simultaneously optimizing more than three objective functions are known as *Many-Objective Optimization Problems* (MaOPs) [11].

Several challenges should be considered when solving problems with more than three objective functions in Pareto optimization contexts [12]. These challenges are intrinsically related to the fact that as the number of objective functions increase, the proportion of non-dominated solutions in the population grows, being increasingly difficult to discriminate among solutions using only the Pareto dominance relation [13]. Additionally, determining which solution to keep and which to discard in order to converge toward the Pareto set is still a relevant issue to be addressed [12], making more difficult to solve MaOPs. Clearly, existing difficulties in solving MaOPs explain why *Many-Objective Optimization* was considered an unexplored domain in resource management of cloud computing datacenters before this research work [14] and no many-objective formulation was proposed for the VMP problem [8, 9, 10].

In this context, research on VMP problems was explored, analyzed and classified to gain a systematic understanding of existing approaches. Consequently, the main scope of the research summarized in this paper is studying first *Many-Objective Virtual Machine Placement* (MaVMP) problems from the perspective of cloud computing providers for several variants of the VMP problems. Different MaVMP problem formulations were proposed for: (1) initial placement of VMs (static) [15, 16], (2) reconfiguration of VMs (semi-dynamic) [17, 18], and (3) cloud computing under uncertainty (dynamic) [19, 20]. Considering the novelty of the proposed formulations, several methods and algorithms were also proposed to address identified issues on solving each studied MaVMP problem.

Additionally, advances of this research are also presented in this paper, such as: experimental comparisons of proposed algorithms against well-known resource allocation algorithms of the industry (e.g. OpenStack), identifying promising results in order to advance the conceptualization of a good tool for resource allocation in real world cloud computing markets. Finally, open challenges are also discussed to guide further advance of the considered cloud computing datacenters research area.

The remainder of this paper is structured as follows: Section 2 presents the proposed taxonomies of the VMP problem for cloud computing datacenters. Section 3 presents concepts on Multi-Objective Optimization, while Section 4 proposes a general optimization framework for MaVMP problems for initial placement of VMs, including a method for effectively reducing the potentially unmanageable number of non-dominated solutions as well as an interactive *Memetic Algorithm* (MA) for solving the formulated problems. Section 5 presents a first formulation of a MaVMP problem with reconfiguration of VMs, considering the simultaneous optimization of five objective functions, as well as an evaluation of five different strategies for automatically selecting a convenient solution from a Pareto set approximation and an extended MA for solving the formulated problem. Section 6 presents a first formulation of a MaVMP problem for cloud computing considering the optimization of four objective functions in a complex IaaS environment, a first scenario-based uncertainty approach for modeling four uncertain parameters of the proposed complex IaaS environment and novel methods and algorithms related to the resolution of the proposed problem in cloud computing datacenters. Finally, Section 7 summarizes the main findings of this research and presents future directions.

2 VMP Taxonomies for Cloud Computing

The VMP problem has been extensively studied in the literature and several surveys have already been presented. Existing surveys focused on specific issues such as: (1) energy-efficient techniques applied to the problem [6, 21], (2) deployment architectures where the VMP problem is applied, as federated clouds [22], and (3) methods for comparing performance of placement algorithms in large on-demand clouds [23].

The above mentioned surveys and research articles focused into specific issues related to the VMP problem. Consequently, López-Pires and Barán proposed in [8, 9, 10] a general and extensive study of a large part of the VMP literature including 84 studied articles [24], presenting a wide analysis of the existing approaches for the formulation and resolution of the VMP as an optimization problem. Additionally, a novel taxonomy was proposed in [8] for the classification of the studied articles by the following criteria: (1) optimization approach, (2) objective function and (3) solution technique.

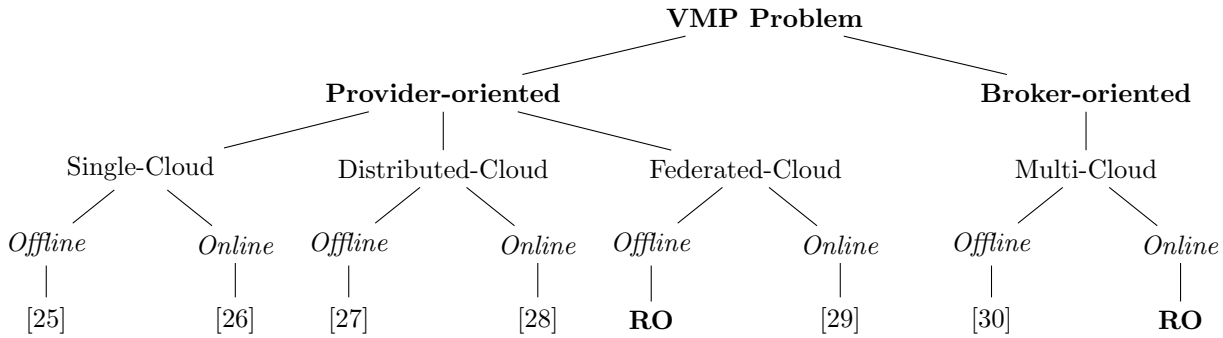

Figure 1: VMP Environment Taxonomy from [9]. Example references for each environment are presented. Unexplored environments are considered as *Research Opportunities* (RO).

The taxonomy presented in [8] was extended in later works [9, 10], including novel taxonomies and presenting a detailed view of the existing approaches as well as several possible research opportunities to further advance in this research area. The taxonomies presented in [9, 10] could guide interested readers to: (1) understand different possible environments where a VMP problem could be studied, considering both provider and broker perspectives in different deployment architectures, (2) identify existing approaches for the formulation and resolution of the VMP as an optimization problem and (3) present a detailed view of the VMP problem, identifying research opportunities to further advance in this research area.

2.1 VMP Environment Taxonomy

Depending on the particular environment where a VMP problem is studied, several different characteristics should be taken into account before considering a particular formulation or technique for the resolution of the considered VMP problem. Here, different possible environments could be identified by classifying research articles in the VMP literature by: (1) orientation, (2) deployment architecture and (3) type of formulation.

For a complete understanding of the possible environments where a VMP problem could be studied, considering both provider and broker perspectives in different deployment architectures and types of formulations, Figure 2.1 presents the taxonomy described in this section including example references from the studied VMP literature [24]. A VMP problem could be studied from both provider or broker perspectives.

A provider-oriented VMP problem could be studied considering one of the following deployment architectures: single-cloud, distributed-cloud or federated-cloud, while a broker-oriented VMP problem could be studied considering a multi-cloud deployment architecture. Additionally, both provider-oriented and broker-oriented VMP problems, in any of the possible deployment architectures, could be studied considering two different types of formulation: offline or online formulations. Interested readers could refer to [9, 10] for details on the VMP Environment Taxonomy presented in Figure 2.1.

2.2 VMP Formulation Taxonomy

Considering each possible environment where a VMP problem could be studied (see Figure 2.1), several different formulations of the problem could be proposed. In this context, formulations of a VMP problem may be classified by the: (1) optimization approach, (2) objective function and (3) solution technique, as initially considered in [8]. A VMP problem could be formulated considering one of the following optimization approaches: (1) mono-objective (MOP), (2) multi-objective solved as mono-objective (MAM) or (3) pure multi-objective (PMO). Once the optimization approach is defined, formulations may also be classified by the objective function(s) studied, both in minimization and maximization contexts. These objective functions could be optimized separately or simultaneously, depending on the selected optimization approach. Finally, solution techniques for solving a VMP problem could be used as a third classification criterion [8, 9, 10].

3 Concepts on Multi-Objective Optimization

A Pure Multi-Objective Optimization (PMO) problem includes a set of p decision variables, q objective functions, and b constraints. Objective functions and constraints are functions of decision variables. In a PMO formulation, x represents the decision vector (or solution), while y represents the objective vector (or solution cost). The decision space is denoted by X, the objective space as Y and can be expressed as [55]:

Table 1: VMP Formulation Taxonomy from [9]. Example references for each environment are presented. Unexplored environments are considered as *Research Opportunities* (RO). In this table $f_1(x)$: Energy Consumption, $f_2(x)$: Network Traffic, $f_3(x)$: Economical Costs, $f_4(x)$: Performance, $f_5(x)$: Resource Utilization.

Technique	**Approach**	**Objective Functions**				
		$\boldsymbol{f_1(x)}$	$\boldsymbol{f_2(x)}$	$\boldsymbol{f_3(x)}$	$\boldsymbol{f_4(x)}$	$\boldsymbol{f_5(x)}$
Optimal Algorithms	MOP	[31]	[32]	[33]	[28]	[34]
	MAM	[5, 35]	[5, 36]	**RO**	**RO**	[35, 36]
	PMO	**RO**	**RO**	**RO**	**RO**	**RO**
Heuristics	MOP	[29]	[37]	[38]	[39]	[34]
	MAM	[40, 26]	[40, 41]	[41, 42]	[26, 43]	[44, 45]
	PMO	**RO**	**RO**	**RO**	**RO**	**RO**
Meta-Heuristics	MOP	[46]	**RO**	[30]	[47]	**RO**
	MAM	[27, 48]	[49, 27]	[50, 48]	[49]	[50, 48]
	PMO	[25, 51]	[25, 52]	[25, 52]	**RO**	[51]
Approximation Algorithms	MOP	[53]	**RO**	**RO**	**RO**	**RO**
	MAM	[54]	**RO**	**RO**	**RO**	**RO**
	PMO	**RO**	**RO**	**RO**	**RO**	**RO**

Optimize:

$$y = f(x) = [f_1(x), f_2(x), ..., f_q(x)] \tag{1}$$

subject to:

$$e(x) = [e_1(x), e_2(x), ..., e_b(x)] \geq 0 \tag{2}$$

where:

$$x = [x_1, x_2, ..., x_p] \in X \tag{3}$$

$$y = [y_1, y_2, ..., y_q] \in Y \tag{4}$$

The set of constrains $e(x) \geq 0$ defines the set of feasible solutions $X_f \subset X$ and its corresponding set of feasible objective vectors $Y_f \subset Y$. The feasible decision space X_f is the set of all decision vectors x in the decision space X that satisfies the constraints $e(x)$ given by (2). The feasible objective space Y_f is the set of the objective vectors y that represents the image of X_f onto Y. These feasible spaces are defined as:

$$X_f = \{x \mid x \in X \wedge e(x) \geq 0\} \tag{5}$$

$$Y_f = \{y \mid y = f(x) \quad \forall x \in X_f\} \tag{6}$$

To compare two solutions in a pure multi-objective context, the concept of Pareto dominance is used. Given two feasible solutions u, $v \in X_f$, u dominates v, denoted as $u \succ v$, if $f(u)$ is better or equal to $f(v)$ in every objective function and strictly better in at least one objective function. If neither u dominates v, nor v dominates u, u and v are said to be non-comparable (denoted as $u \sim v$).

A decision vector x is non-dominated with respect to a set U, if there is no member of U that dominates x. The set of non-dominated solutions of the whole set of feasible solutions X_f, is known as optimal Pareto set P^*. The corresponding set of objective vectors constitutes the optimal Pareto front PF^*.

Taking into account the large number of existing objective functions and possible approaches for objective function modeling identified in [24, 8], PMO approaches could result in more realistic formulations of a VMP problem, optimizing more than just one objective function at a time (e.g. maximizing economical revenue by simultaneously optimizing economical penalties for SLA violations, operational costs and profit for leasing computational resources).

4 MaVMP for Initial Placement

As previously presented, provider-oriented VMP problems considering PMO optimization represent a research challenge for resource allocation in cloud computing datacenters. It is worth remembering that no many-objective formulation had already been proposed for the VMP problem (MaVMP) in the specialized literature [8] before our work was published in [15].

4.1 Many-Objective Optimization Framework

This section summarizes a general many-objective optimization framework, which is able to consider as many objective functions as needed when solving MaVMP problems for initial placement. As an example of utilization of the proposed framework, a formulation of a MaVMP problem is proposed, considering the simultaneous optimization of the following five objective functions: (1) power consumption, (2) network traffic, (3) economical revenue, (4) *quality of service* (QoS) and (5) network load balancing. In the presented MaVMP formulation, a multi-level priority is associated to each VM, representing a *Service Level Agreement* (SLA) to be considered in the placement process in order to effectively prioritize important VMs.

Formally, the proposed offline (static) MaVMP problem for initial placement can be enunciated as: *Given a set of PMs, $H = \{H_1, H_2, ..., H_n\}$, a network topology G and a set of VMs, $V = \{V_1, V_2, ..., V_m\}$, it is sought a correct placement of the set of VMs V into the set of PMs H satisfying the b constraints of the problem and simultaneously optimizing all q objective functions defined in this formulation (as energy consumption, network traffic, economical revenue, QoS and load balancing in the network), in a pure many-objective context.*

Due to space limitations, the complete mathematical formulation as well as other details are not included in this summary. Interested readers could refer to [15] and [16] for details on this VMP variant.

As a relevant issue to be solved when considering this particular MaVMP, the general many-objective optimization framework for the VMP problem proposed in this dissertation considers that as the number of conflicting objectives of a MaVMP problem formulation increases, the total number of non-dominated solutions increases (even exponentially in some cases), being increasingly difficult to discriminate among solutions using only the dominance relation [12]. For this reason, this work proposes the utilization of lower and upper bounds associated to each objective function $f_z(x)$, where $z \in \{1, \ldots, q\}$ ($L_z \leq f_z(x) \leq U_z$), to be able to reduce iteratively the number of non-dominated solutions of the known approximation to the Pareto set (P_{known}).

Based on many objective functions and constraints detailed in [16], a MaVMP formulation may be expressed as:
Optimize:

$$y = f(x) = [f_1(x), f_2(x), f_3(x), \ldots, f_q(x)], \quad q > 3 \tag{7}$$

where for example:

$$\begin{aligned} f_1(x) &= \text{power consumption;} \\ f_2(x) &= \text{network traffic;} \\ f_3(x) &= \text{economical revenue;} \\ f_4(x) &= \text{quality of service;} \\ f_5(x) &= \text{network load balancing;} \\ &\vdots \\ f_q(x) &= \text{any other considered objective function.} \end{aligned} \tag{8}$$

subject to constraints as:

$$\begin{aligned} e_1(x) &: \text{unique placement of VMs;} \\ e_2(x) &: \text{assure provisioning of highest SLA;} \\ e_3(x) &: \text{processing resource capacity of PMs;} \\ e_4(x) &: \text{memory resource capacity of PMs;} \\ e_5(x) &: \text{storage resource capacity of PMs;} \\ e_6(x) &: f_1(x) \in [L_1, U_1]; \\ e_7(x) &: f_2(x) \in [L_2, U_2]; \\ e_8(x) &: f_3(x) \in [L_3, U_3]; \\ e_9(x) &: f_4(x) \in [L_4, U_4]; \\ e_{10}(x) &: f_5(x) \in [L_5, U_5]; \\ &\vdots \\ e_r(x) &: \text{any other considered constraint.} \end{aligned} \tag{9}$$

To solve the formulated MaVMP problem, an interactive *Memetic Algorithm* (MA) is proposed considering particular challenges associated to the resolution of a VMP problem for initial placement in a many-objective context.

4.2 Interactive Memetic Algorithm for MaVMP

A *Memetic Algorithm* (MA) could be understood as an *Evolutionary Algorithm* (EA) that in addition to the standard selection, crossover and mutation operators of most *Genetic Algorithms* (GAs) includes a local optimization operator to obtain good solutions even at early generations [56]. In the studied VMP context, it is valuable to obtain good quality solutions in a short time. Consequently, a MA could be considered as a promising solution technique for VMP problems.

An interactive MA is presented for solving the proposed MaVMP problem for initial placement, to simultaneously optimize the following objective functions: (1) power consumption, (2) network traffic, (3) economical revenue, (4) *quality of service* (QoS) and (5) network load balancing. The proposed algorithm is extensible to consider as many objective functions as needed while only minor modifications may be needed in the code if objective functions change.

It was shown in [57] that many-objective optimization using *Multi-Objective Evolutionary Algorithms* (MOEAs) is an active research area, having multiple challenges that need to be addressed. The interactive MA presented in this section is a viable way to solve a MaVMP problem for initial placement, including desirable ranges of values for the objective function costs in order to interactively control the possible huge number of feasible non-dominated solutions, as described in Section 3. The proposed interactive MA presented in Algorithm 1 basically works as follows:

At step 1, it is verified if the problem has at least one solution to continue with next steps. If there is no possible solution to the problem, the algorithm returns an appropriate error message. If the problem has at least one solution, the algorithm proceeds to step 2, which generates a set of random candidates Pop_0, whose solutions are repaired at step 3 to ensure that Pop_0 contains only feasible solutions. Then, the algorithm tries to improve candidates at step 4 using local search. With the obtained non-dominated solutions, the first Pareto set approximation P_{known} is generated at step 5. After initialization at step 6, evolution begins (iterations between steps 7 and 18). The evolutionary process basically follows the same behaviour: solutions are selected from the union of P_{known} with the evolutionary set of solutions (or population) also known as Pop_t (step 8), crossover and mutation operators are applied as usual (step 9), and eventually solutions are repaired, as there may be infeasible solutions (step 10). Improvements of solutions of the evolutionary population Pop_t may be generated at step 11 using local search (local optimization operators). At step 12, the Pareto set approximation P_{known} is updated (if applicable); while at step 13 the generation (or iteration) counter is updated. At step 15 the decision maker adjust the lower and upper bounds if it is necessary, while at step 17 a new evolutionary population Pop_t is selected. The evolutionary process is repeated until the algorithm meets a stopping criterion (such as maximum number of generations), finally returning the set of non-dominated solutions P_{known} at step 19.

Due to space limitations, details on population initialization, solution reparation, local search, fitness function and variation operators are not included in this summary.

Algorithm 1: Interactive Memetic Algorithm from [17].

Data: datacenter infrastructure
Result: Pareto set approximation P_{known}
1 check if the problem has a solution
2 initialize set of solutions Pop_0
3 Pop'_0 = repair infeasible solutions of Pop_0
4 Pop''_0 = apply local search to solutions of Pop'_0
5 update set of solutions P_{known} from Pop''_0
6 $t = 0; Pop_t = Pop''_0$
7 **while** *stopping criterion is not met* **do**
8 Q_t = selection of solutions from $Pop_t \cup P_{known}$
9 Q'_t = crossover and mutation of solutions of Q_t
10 Q''_t = repair infeasible solutions of Q'_t
11 Q'''_t = apply local search to solutions of Q''_t
12 update set of solutions P_{known} from Q'''_t
13 increment t
14 **if** *interaction is needed* **then**
15 ask for decision maker modification of (L_z and U_z)
16 **end**
17 Pop_t = non-dominated sorting from $Pop_t \cup Q'''_t$
18 **end**
19 **return** Pareto set approximation P_{known}

Given that the number of non-dominated solutions may rapidly increase, an interactive approach is recommended. That way, a decision maker can introduce new constraints or adjust existing ones, while the execution continues, learning about the shape of the Pareto front in the process. The present work considers lower and upper bounds associated to each objective function in order to help the decision maker to reduce interactively the potential huge number of solutions in the Pareto set approximation P_{known}, while observing the evolution of its corresponding Pareto front PF_{known} to the region of his preference.

4.3 Experimental Results: Interactive Bounds

Several experimental evaluations were performed in order to validate the good quality of solutions obtained by the proposed interactive MA against optimal solutions when possible, as well as its scalability for solving problem instances with large number of PMs and VMs [15, 16]. This section focuses on summarizing experimental results related to the evaluation of the lower and upper bounds proposed to address issues associated to MaVMP problems for initial placement previously described.

For example, a problem instance composed by 12 PMs and 50 VMs, one run of the proposed algorithm was completed, after evolving populations of 100 individuals for 300 generations. The number of generations was incremented for this experiment from 100 to 300, taking into account the large number of possible solutions for the particular problem considered. An interactive adjustment of the lower or upper bounds associated to each objective function was performed after every 100 generations in order to converge to a treatable number of solutions. It is important to remark that the interactive adjustment used in this experiment is only one of several possible ones.

As an example, we may consider: (1) automatically adjusting a % of the lower bounds associated to maximization objective functions when the Pareto front has a defined number of elements or (2) manually adjusting upper bounds associated to minimization objective functions until the Pareto front does not have more than 20 elements, just to cite a pair of alternatives.

The Pareto front approximation PF_{known} represents the complete set of Pareto solutions considering unrestricted bounds ($L_z = -\infty$ and $U_z = \infty$). On the other hand, Pareto front approximation $PF_{reduced}$ represents the reduced set of Pareto solutions obtained by interactively adjusting bounds L_z and U_z. After 100 generations, the proposed algorithm obtained 251 solutions with unrestricted bounds.

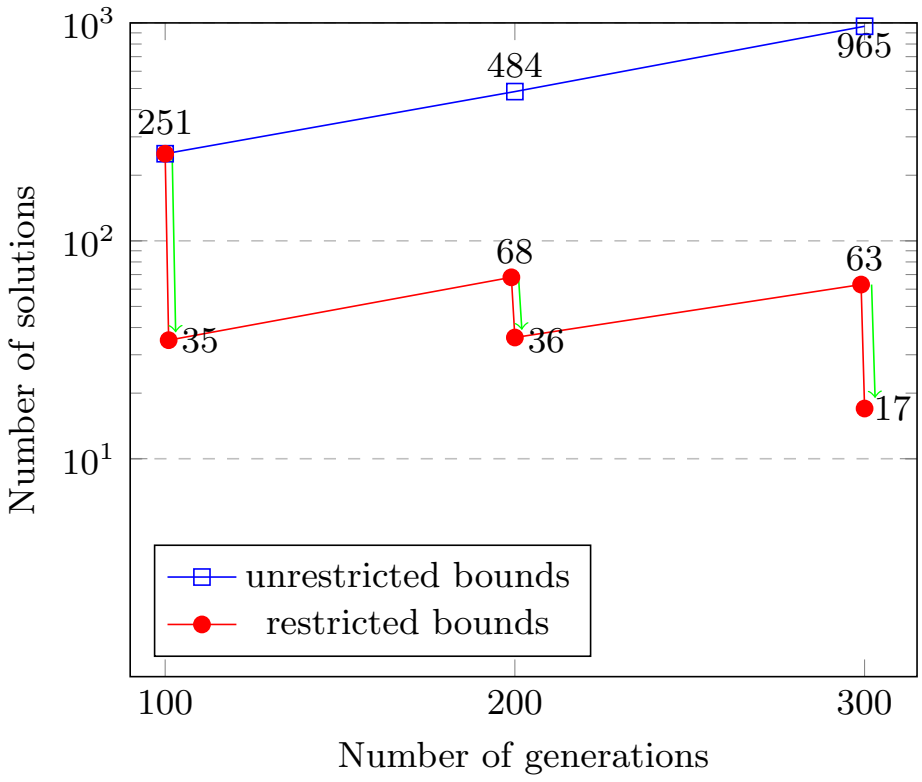

Figure 2: Summary of results obtained in [16] using restricted lower and upper bounds against unrestricted bounds.

A decision maker evaluated the bounds associated to $f_1(x)$ (power consumption) and adjusted the upper bound U_1 to $U_1' = 9000$ [W], selecting only 35 out of the 251 solutions (not considering 216 otherwise feasible solutions) for the $PF_{reduced}$ as shown in Figure 2. After 200 generations, the algorithm obtained a total of 484 solutions with unrestricted bounds. Considering instead $U_1' = 9000$ [W], the algorithm only found 68 solutions. The decision maker evaluated the bounds associated to $f_2(x)$ (network traffic) and adjusted the upper bound U_2 to $U_2' = 115$ [Mbps], selecting only 36 out of the 68 solutions for the $PF_{reduced}$. After 300 generations, the algorithm obtained a total of 965 solutions with unrestricted bounds. Considering $U_1' = 9000$ [W] and $U_2' = 115$ [Mbps], the algorithm found 63 solutions. The decision maker evaluated the bounds associated to $f_3(x)$ (economical revenue) and adjusted the lower bound L_3 to $L_3' = 13500$ [$], selecting only 17 out of the 63 solutions for the final $PF_{reduced}$ as shown in Figure 2.

Clearly, at the end of the iterative process, the decision maker found 17 solutions according to his preferences instead of 965 unmanageable candidate solutions.

5 MaVMP with Reconfiguration of VMs

Once an initial placement of VMs have been performed, a virtualized datacenter could be reconfigured through VM live migration in order to maintain efficiency in operations, considering that the set of requested VMs changes over time. This particular semi-dynamic formulation of a VMP problem may be studied as an approximation to dynamic formulations of real-world IaaS environments.

5.1 Problem Formulation

According to [8, 9], the optimization of power consumption is the most studied objective function in VMP literature [35, 6]. Furthermore, network traffic [5] and economical revenue [58, 26] are also very studied as objective functions for the VMP problem. For a VMP problem formulation with reconfiguration of VMs, two additional objective functions associated to migration of VMs represent challenges for CSPs: number of VM migrations [59] and network traffic overhead for VM migrations [7].

Considering the large number of existing objective functions for VMP problems identified in [8, 9], Lopez-Pires and Baran have proposed in [15, 16] a many-objective optimization framework in order to consider as many objective functions as needed when solving a MaVMP problem for initial placement. To the best of the authors' knowledge there was no published work presenting a formulation of a MaVMP with reconfiguration of VMs before our work in [17]. This section extends formulations presented in [15, 16], proposing for the first time a MaVMP with reconfiguration of VMs, considering this time the simultaneous optimization of the following five objective functions: (1) power consumption, (2) inter-VM network traffic, (3) economical revenue, (4) number of VM migrations and (5) network traffic overhead for VM migrations.

Formally, the proposed offline (semi-dynamic) MaVMP problem with reconfiguration of VMs can be enunciated as:

Given the available PMs and their specifications, the requested VMs and their specifications, the network traffic between VMs and the current placement of the VMs, it is sought a new placement of the set of VMs in the set of PMs, satisfying the constraints of the problem while simultaneously optimizing all defined objective functions (as power consumption, inter-VM network traffic, economical revenue, number of VM migrations and network traffic overhead for VM migration), in a pure many-objective context, before selecting a specific solution for a given time instant t.

Due to space limitations, the complete mathematical formulation as well as other details are not included in this summary. Interested readers could refer to [15] and [16] for details on this VMP variant.

To solve the formulated MaVMP problem, the interactive MA presented in Section 4.2 was extended to consider challenges associated to solve a MaVMP problem with reconfigurations of VMs, as next introduced.

5.2 Extended Memetic Algorithm for MaVMP

Several challenges need to be addressed for MaVMP problems with reconfiguration of VMs. In Pareto-based algorithms, the Pareto set approximation can include a large number of non-dominated solutions. Selecting one of these solutions can be considered a relevant issue in semi-dynamic environments. In consequence, this work evaluates the following selection strategies: (1) random, (2) preferred solution [57], (3) minimum distance to origin, (4) lexicographic order (provider preference) and (5) lexicographic order (service preference) to identify convenient strategies for automatic selection of a solution for the considered problem.

For a MaVMP problem with reconfiguration of VMs, at each time instant the set of feasible placements can be composed by a large number of non-dominated solutions. Therefore, the proposed algorithm automatically selects one of the possible placements after each time instant according to one of the considered selection strategies.

Table 2: Selection Strategy Comparison from [17]. In this table $f_1(x)$: Power Consumption, $f_2(x)$: Inter-VM Network Traffic, $f_3(x)$: Economical Revenue, $f_4(x)$: Number of VM Migrations, $f_5(x)$: Network Traffic overhead for VM Migrations.

Selection Strategy	Objective Functions Averages					Dominance ($row \succ column$)					Preference ($row \succ column$)				
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$S1$	$S2$	$S3$	$S4$	$S5$	$S1$	$S2$	$S3$	$S4$	$S5$
$\boldsymbol{W_1}$															
$S1$	9,908	19,981	32,623	44	1,526										
$S2$	9,827	19,991	32,623	8	180						$\succ$			$\succ$	
$S3$	9,639	19,228	32,623	6	124	$\succ$	$\succ$				$\succ$	$\succ$		$\succ$	$\succ$
$S4$	8,543	21,038	32,623	19	520						$\succ$				
$S5$	10,395	21,957	32,623	5	150										
$\boldsymbol{W_2}$															
$S1$	104,559	371,664	325,217	650	26,886										
$S2$	104,835	373,467	325,217	37	1,204						$\succ$			$\succ$	
$S3$	104,378	370,489	325,217	26	804	$\succ$	$\succ$				$\succ$	$\succ$		$\succ$	
$S4$	103,175	374,210	325,217	92	3,531						$\succ$				
$S5$	104,860	373,230	325,217	20	618							$\succ$		$\succ$	

Considering that the MA proposed for this particular MaVMP problem is an extension of the one presented in Section 4.2 and due to space limitations, details are not included in this summary. Nevertheless, interested readers may refer to [17] for a more detailed explanation.

5.3 Experimental Results: Strategies Evaluation

Table 2 summarizes the results obtained considering two different workloads. First workload W_1 is composed by 10 PMs, running in average 100 VMs for 24 time instants, while the second workload W_2 is composed by 100 PMs, running in average 1000 VMs also for 24 time instants.

As expected, when the lexicographic order is used, the most important objective function is the one with the best results, i.e. the $S4$ strategy (provider preference) obtains the best results in power consumption $f_1(x)$, with 20% less power consumption than the worst strategy in W_1 and 2% less power consumption than the worst strategy in W_2. When service perspective is prioritized ($S5$), the objective functions $f_4(x)$ and $f_5(x)$ obtain the best results.

However, as the focus of this work is the simultaneous optimization of all five objective functions with a PMO optimization approach, a comparison is made considering the concept of Pareto dominance. As seen in Table 2 (dominance column), the $S3$ strategy dominates $S2$ and $S1$ in both experiments; however, it is non-comparable with respect to $S4$ and $S5$ in both presented problem instances.

Given that $S3$ cannot be declared as the best strategy considering exclusively Pareto dominance, a further comparison of selection strategies using *preference* (i.e. larger number of better objective functions) criteria [57] is presented in the corresponding column of Table 2.

It may seem intuitive that the $S2$ strategy (that uses the preference criterion) should be the best; however, Table 2 shows that strategy $S3$ is preferred not only to $S2$ but also to $S1$ and to $S4$ in both problem instances.

Additionally, it can be seen that $S3$ is preferred to $S5$ in problem instance W_1 while no strategy is preferred to $S3$, indicating that $S3$ (distance to origin) could be considered the preferred (best) strategy for solving the proposed MaVMP problem with reconfiguration of VMs.

As a consequence of the above results, for production cloud datacenters, instead of calculating all the Pareto set approximation, the $S3$ strategy (distance to origin) could be used to combine all considered objective functions into only one objective function, therefore solving the studied problem considering a Multi-Objective problem solved as Mono-Objective (MAM) approach.

6 Uncertain MaVMP for Cloud Computing

This section proposes a complex IaaS environment for VMP problems, considering service elasticity and overbooking of physical resources. To the best of the authors' knowledge, there is no previous published work considering these fundamental criteria, directly related to the most relevant dynamic parameters in the specialized literature [60]. In order to model this complex IaaS environment for VMP problems, cloud services (i.e. a set of inter-related VMs) are considered instead of only isolated VMs.

It is worth remembering that VMP is a NP-Hard combinatorial optimization problem [61]. From an IaaS provider perspective, it is mostly formulated as an online problem and must be solved with short time constraints [8]. Online decisions made along the operation of a dynamic cloud computing infrastructure negatively affects the quality of solutions in VMP problems when comparing to offline decisions, studied as part of this dissertation in [19]. In this context, offline algorithms present a substantial advantage over online alternatives. Unfortunately, offline formulations are not appropriate for highly dynamic environments for real-world IaaS providers, where cloud services are requested dynamically according to current demand.

In what follows, this section presents a two-phase optimization scheme, decomposing the VMP problem into two different sub-problems, combining advantages of online and offline VMP formulations considering a complex IaaS environment: (1) *incremental VMP* (iVMP) and (2) *VMP reconfiguration* (VMPr). This two-phase optimization scheme combines both online (iVMP) and offline (VMPr) algorithms for solving each considered VMP sub-problem.

In online algorithms for solving the proposed problem, placement decisions are performed at each discrete time t. The formulation of the proposed iVMP (online) problem is based on [19], formally enunciated as: *Given a complex IaaS environment composed by a set of PMs (H), a set of active VMs already requested before time t ($V(t)$), and the current placement of VMs into PMs (i.e. $x(t)$), it is sought an incremental placement of $V(t)$ into H for the discrete time $t+1$ ($x(t+1)$) without migrations, satisfying the problem constraints and optimizing the considered objective functions.*

On the contrary, offline algorithms solve a VMP problem considering a static environment where VM requests do not change over time and considers migration of VMs between PMs. The formulation of the proposed VMPr (offline) problem is based on [17, 15] and could be stated as:

Given a current placement of VMs into PMs ($x(t)$), it is sought a placement reconfiguration through migration of VMs between PMs for a discrete time t (i.e. $x'(t)$), satisfying the constraints and optimizing the considered objectives.

For IaaS customers, cloud computing resources often appear to be unlimited and can be provisioned in any quantity at any required time [2]. This formulation considers a basic federated-cloud deployment architecture for the problem.

It is important to remember that more than 60 different objective functions have been proposed for VMP problems [8]. In this context, the number of considered objective functions may rapidly increase once a complete understanding of the VMP problem is accomplished for practical problems, where several different parameters should be ideally taken into account. Consequently, a formulation of the VMP problem is presented considering the optimization of the following four objective functions: (1) power consumption, (2) economical revenue, (3) resource utilization and (4) reconfiguration time.

Due to the randomness of customer requests, VMP problems should be formulated under uncertainty. This work presents a scenario-based uncertainty approach for modeling uncertain parameters, considering a two-phase optimization scheme for VMP problems in complex IaaS environments.

This work identifies two main research questions related to the considered two-phase optimization scheme:

- ***Research Question 1 (RQ1):*** when or under which circumstances the VMPr phase should be triggered? (*VMPr Triggering* method).
- ***Research Question 2 (RQ2):*** what should be done with cloud service requests arriving during recalculation time in the VMPr phase? (*VMPr Recovering* method).

The presented optimization scheme for the VMP problem introduces novel methods to decide when to trigger placement reconfigurations with migration of VMs between PMs (defined as *VMPr Triggering*) and what to do with cloud services requested during placement recalculation times (defined as *VMPr Recovering*).

6.1 Related Works and Motivation

To the best of the authors' knowledge, there is no published work considering uncertainty of parameters for provider-oriented VMP problem formulations. Consequently, this section mainly focus on describing IaaS environments considered on each related work, as well as already proposed VMPr Triggering and VMPr Recovering methods, when applicable. The most relevant works for this research are briefly described [44, 6] and a summary of considered related works is presented in Table 3.

Calcavecchia et al. studied in [44] a practical model of cloud service placement for a stream (or workload) of requests where inter-related VMs are created and destroyed, considering CPU overbooking and static reservation of VMs resources. The mentioned cloud service placement model is composed by two phases: (1) continuous deployment (or iVMP) and (2) ongoing optimization (or VMPr).

Table 3: Summary of IaaS environments and VMPr methods already studied in related works from [20]. N/A indicates a Not Applicable criterion.

Ref	Overbooking Type	Elasticity Type	VMPr Triggering	VMPr Recovering
[44]	CPU	Not Considered	Periodically	Cancellation
[62]	Not Considered	Not Considered	Periodically	Not Considered
[63]	Not Considered	Not Considered	Periodically	Not Considered
[64]	Not Considered	Not Considered	Periodically	Not Considered
[65]	CPU and RAM	Not Considered	Periodically	Not Considered
[66]	Not Considered	Not Considered	Periodically	Not Considered
[67]	Not Considered	Not Considered	Continuously	Not Considered
[6]	CPU	Not Considered	Threshold-based	N/A
[68]	CPU, RAM, Network	Not Considered	Threshold-based	N/A
[69]	CPU	Horizontal	Threshold-based	N/A
This work	*CPU, RAM, Network*	*Vertical, Horizontal*	*Prediction-based*	*Update-based*

The continuous deployment is performed by a *Best-Fit Decreasing* (BFD) heuristic while a *Backward Speculative Placement* (BSP) is performed in the ongoing optimization phase. To improve a current placement, the ongoing optimization is periodically triggered for the duration of the workload and canceled whenever a new request is received.

Beloglazov et al. identified in [6] two stages for the VMP problem: (1) initial admission of VMs and (2) optimization of the current placement. For the admission of VMs (or iVMP) a *Modified Best-Fit Decreasing* (MBFD) algorithm is considered, using the CPU utilization of VMs to sort a list of VM requests and allocate each VM into a PM that provides the minimum increment in power consumption. Additionally, the optimization of the current placement (or VMPr) is triggered whenever an overloaded or underloaded PM is detected, according to well-defined CPU utilization thresholds. In this case, the VMPr runs distributively for each overloaded or underloaded PM to migrate VMs from overloaded PMs until each PM is appropriately loaded, consolidating VMs from underloaded PMs to decrease the number of running PMs to the minimum possible number. It is important to consider that this threshold-based triggering represents a decentralized decision process, relaxing the computational complexity of the VMP problem. Consequently, it is not necessary to consider the arrival of VM requests during the reconfiguration because no offline centralized decision is performed. Considering the VMPr, a selection process is performed to determinate which VMs should be migrated (all in case of underloaded PMs). Selected VMs are allocated by the MBFD algorithm into PMs considering CPU overbooking.

In summary, as presented in Table 3, most of the related works that consider IaaS environments with overbooking are limited to CPU resources. Only [68] considered overbooking for all available resources, as proposed in this work. Additionally, studied IaaS environments with elasticity are limited to horizontal elasticity [69], while this work considers both vertical and horizontal elasticity.

According to the studied articles (see Table 3), existing VMPr Triggering methods may be classified as: (1) periodical and (2) threshold-based. Periodically triggering the VMPr could present disadvantages when defining a fixed reconfiguration period (e.g. every 10 minutes) because reconfigurations may be required before the established time or in certain cases the reconfiguration may not be necessary. For threshold-based approaches, thresholds are defined in terms of utilization of resources (e.g. CPU) without a complete knowledge of global optimization objectives. Therefore, this work proposes a novel prediction-based approach as VMPr Triggering method, statistically analyzing the objective function costs and proactively detecting requirements for triggering the VMPr.

Additionally, most of the studied works do not consider any VMPr Recovering method, when applicable. Only Calcavecchia et al. studied in [44] a very basic approach, canceling the VMPr whenever a new request is received. Consequently, the VMPr is only performed in periods with no requests, that could result unrealistic, specially for highly loaded IaaS environments. On the other hand, this work proposes a novel VMPr Recovering method based on updating the potentially obsolete placement recalculated in the VMPr phase with the required cloud services created, modified and removed during the recalculation time.

6.2 Problem Formulation

This section presents a formulation of the VMP problem under uncertainty considering a two-phase scheme for the optimization of the following objective functions: (1) power consumption, (2) economical revenue, (3) resource utilization and (4) placement reconfiguration time.

Due to space limitations, the complete mathematical formulation as well as other details are not included in this summary. Interested readers could refer to [20] for details on this particular VMP variant.

According to [8], this section focuses on a provider-oriented VMP for federated-clouds, considering a combination of two types of formulations: (1) online (i.e. iVMP) and (2) offline (i.e. VMPr).

An online problem formulation is considered when inputs of the problem change over time and the algorithm does not have the entire input set available from the start [7]. On the other hand, if inputs of the problem do not change over time, the formulation is considered offline (e.g. MAs proposed in [17] and [15]).

In order to model a dynamic VMP environment taking into account both vertical and horizontal elasticity of cloud services, the set of requested VMs $V(t)$ may include the following types of requests for cloud service placement at each discrete time t:

- **cloud services creation:** where new cloud services S_b, composed by one or more VMs V_j, are created. Consequently, the number of VMs at each discrete time t (i.e. $m(t)$) is a function of time;
- **scale-up / scale-down of VMs resources:** where one or more VMs V_j of a cloud service S_b increases (scale-up) or decreases (scale-down) its capacities of virtual resources with respect to current demand (vertical elasticity). In order to model these considerations, virtual resource capacities of a VM V_j (i.e. $Vr_{1,j}(t)$-$Vr_{3,j}(t)$) are a function of time, as well as the associated economical revenue ($R_j(t)$);
- **cloud services scale-out / scale-in:** where a cloud service S_b increases (scale-out) or decreases (scale-in) the number of associated VMs according to current demand (horizontal elasticity). Consequently, the number of VMs V_j in a cloud service S_b at each discrete time t, denoted as $mS_b(t)$, is a function of time;
- **cloud services destruction:** where virtual resources of cloud services S_b, composed by one or more VMs V_j, are released.

In most situations, virtual resources requested by cloud services are dynamically used, giving space to re-utilization of idle resources that were already reserved. Information about the utilization of virtual resources at each discrete time t is required in order to model a dynamic VMP environment where IaaS providers consider overbooking of both server and networking physical resources.

6.2.1 Normalization and Scalarization Methods

As a consequence of experimental results obtained in a previous work by the authors [17] for VMP problems optimizing multiple objective functions, even in a many-objective optimization context for cloud computing datacenters, instead of calculating a whole Pareto set approximation, a scalarization method (e.g. minimum distance to origin) is suggested to combine all considered objective functions into a single objective function; therefore, solving the studied problem considering a Multi-Objective problem solved as Mono-Objective (MAM) approach [8]. Consequently, each of the considered objective function must be formulated in a single optimization context (in this case, minimization) and each objective function cost must be normalized to be comparable and combinable as a single objective.

This work normalizes each objective function cost by calculating $\hat{f}_i(x,t) \in \mathbb{R}$, where $0 \leq \hat{f}_i(x,t) \leq 1$ for each original objective function $f_i(x,t)$.

$$\hat{f}_i(x,t) = \frac{f_i(x,t) - f_i(x,t)_{min}}{f_i(x,t)_{max} - f_i(x,t)_{min}} \tag{10}$$

where:

$\hat{f}_i(x,t)$:	Normalized cost of objective function $f_i(x,t)$ at instant t;
$f_i(x,t)$:	Cost of original objective function;
$f_i(x,t)_{min}$:	Minimum possible cost for $f_i(x,t)$;
$f_i(x,t)_{max}$:	Maximum possible cost for $f_i(x,t)$.

Finally, the presented normalized objective functions are combined into a single objective considering a minimum Euclidean distance to the origin, expressed as:

$$F(x,t) = \sqrt{\sum_{i=1}^{q} \hat{f}_i(x,t)^2} \tag{11}$$

Table 4: Summary of evaluated algorithms as well as their corresponding VMPr Triggering and Recovering methods. N/A indicates a Not Applicable criterion.

Algorithm	**Decision Approach**	**iVMP**	**VMPr**	**VMPr Triggering**	**VMPr Recovering**
A0 - inspired in [71]	N/A	FFD	N/A	N/A	N/A
A1 - inspired in [44]	Centralized	FFD	MA	Periodically	Cancellation
A2 - inspired in [6]	Distributed	FFD	MMT	Threshold-based	N/A
A3 - from [20]	**Centralized**	**FFD**	**MA**	**Prediction-based**	**Update-based**
A4 - inspired in [72]	N/A	OpenStack	N/A	N/A	N/A

where:

$F(x,t)$: Single objective function combining each $\hat{f}_i(x,t)$ at instant t;
$\hat{f}_i(x,t)$: Normalized cost of objective function $f_i(x,t)$ at instant t;
q: Number of objectives. In this work $q = 4$.

6.2.2 Scenario-based Uncertainty Modeling

In this work, uncertainty is modeled through a finite set of well-defined scenarios S [70], where the following uncertain parameters are considered: (1) virtual resources capacities (vertical elasticity), (2) number of VMs that compose cloud services (horizontal elasticity), (3) utilization of CPU and RAM memory virtual resources and (4) utilization of networking virtual resources (both relevant for overbooking).

For each scenario $s \in S$, a temporal average value of the objective function $F(x,t)$ presented in Equation (11) is calculated as:

$$\overline{f_s(x,t)} = \frac{\sum_{t=1}^{t_{max}} F(x,t)}{t_{max}} \tag{12}$$

where:

$\overline{f_s(x,t)}$: Temporal average of combined objective function for all discrete time instants t in scenario $s \in S$;
t_{max}: Duration of a scenario (or simulation) in discrete time instants.

As previously described, when parameters are uncertain, it is important to find solutions that are acceptable for any (or most) considered scenario $s \in S$. This work considers minimization of the following criteria to select among solutions from different evaluated alternatives as: (1) average [70], (2) maximum [70] and (3) minimum objective function costs:

$$F_1 = \overline{F(x,t)} = \frac{\sum_{s=1}^{|S|} \overline{f_s(x,t)}}{|S|} \tag{13}$$

$$F_2 = \max_{s \in S}(\overline{f_s(x,t)}) \tag{14}$$

$$F_3 = \min_{s \in S}(\overline{f_s(x,t)}) \tag{15}$$

where:

F_1: Average $\overline{f_s(x,t)}$ for all scenarios $s \in S$;
F_2: Maximum $\overline{f_s(x,t)}$ considering all scenarios $s \in S$;
F_3: Minimum $\overline{f_s(x,t)}$ considering all scenarios $s \in S$.

Although F_1 and F_2 are the most studied criteria in the specialized literature [70], this work also considers F_3 as a criterion just to demonstrate that experimental conclusions do not change when also considering minimum costs.

6.3 Evaluated Algorithms

Taking into account that this work presents a novel uncertain VMP formulation considering a complex IaaS environment [20], there is no published alternatives to which we can compare the proposed algorithm. Therefore, the main goal of the experimental evaluation to be presented is to validate that the proposed VMPr Triggering and VMPr Recovering methods improve the quality of solutions, against adapted state-of-the-art alternatives that originally consider only partially the proposed complex IaaS environment.

This work firstly evaluates four different algorithms (*A0* to *A3* in Section 6.4), presented in Table 4. First, Algorithm 0 (*A0*) is evaluated considering only the online iVMP phase, without taking into account reconfiguration of VMs. Algorithm 1 (*A1*) is inspired in [44], considering a centralized decision approach while Algorithm 2 (*A2*) is inspired in [6] following a distributed decision approach. Additionally, Algorithm 3 (*A3*) considers a centralized decision approach implementing the prediction-based VMPr Triggering and update-based VMPr Recovering methods proposed in [20]. In this context, *A1* and *A2* consider original VMPr Triggering and VMPr Recovering methods proposed on each research work [44, 6].

Additionally, this work presents a comparison of two algoritms (*A3* and *A4* in Section 6.5), presented in Table 4. Here, *A4* is inspired in [72], representing main functioning of industry de-facto standard OpenStack.

6.3.1 Proposed Prediction-based Triggering

Considering the main identified issues related to the studied VMPr Triggering methods, this work presents a prediction-based VMPr Triggering method from [20], statistically analyzing the global objective function $F(x,t)$ that is optimized (see Equation (11)) and proactively detecting situations where a VMPr triggering is potentially required for a placement reconfiguration.

The presented prediction-based VMPr Triggering method considers *Double Exponential Smoothing* (DES) [73] as a statistical technique for predicting values of the objective function $F(x,t)$, formulated next in Equations (16) to (18):

$$S_t = \alpha \times Z_t + (1-\tau)(S_{t-1} + b_{t-1}) \tag{16}$$

$$b_t = \tau(S_t - S_{t-1}) + (1-\tau)(b_{t-1}) \tag{17}$$

$$\overline{Z}_{t+1} = S_t + b_t \tag{18}$$

where:

α:	Smoothing factor, where $0 \leq \alpha \leq 1$;
τ:	Trend factor, where $0 \leq \tau \leq 1$;
Z_t:	Known value of $F(x,t)$ at discrete time t;
S_t:	Expected value of $F(x,t)$ at discrete time t;
b_t:	Trend of $F(x,t)$ at discrete time t;
$\overline{Z}_{t+1}$:	Value of $F(x,t+1)$ predicted at discrete time t.

At each discrete time t, the proposed prediction-based VMPr Triggering method predicts the next $\hat{N}$ values of $F(x,t)$ and effectively triggers the VMPr phase in case $F(x,t)$ is predicted to consistently increase, considering that $F(x,t)$ is being minimized.

6.3.2 Proposed Update-based Recovering

Considering an identified opportunity to improve the existing VMPr Recovering method [44], this work proposes a novel update-based VMPr Recovering method based on updating the placement reconfiguration calculated in the VMPr phase, according to changes that happened during the placement recalculation time, applying operations to update the potentially obsolete placement. Details can be seen in [20].

6.4 Experimental Results: VMPr Triggering and Recovering Methods

Table 5 presents values of the considered evaluation criteria, i.e. F_1, F_2 and F_3 costs (see Equation (13) to (15)), summarizing results obtained in performed simulations. The mentioned evaluation criteria are presented separately for each of the five considered IaaS cloud datacenter. It is worth noting that the considered IaaS cloud datacenters represent datacenters of different sizes and consequently, the considered workload traces represent different load of requested CPU resources (e.g. Low ($\leq 30\%$), Medium ($\leq 60\%$), High ($\leq 90\%$), Full ($\leq 98\%$) and Saturate ($\leq 120\%$)) workloads. The main idea of evaluating different load of requested CPU resources is inspired in [74].

Table 5: Summary of evaluation criteria in experimental results for evaluated algorithms from [20].

Criterion	⟍	DC_1	DC_2	DC_3	DC_4	DC_5	*Rank*
F_1	*A0*	0.691	0.758	0.855	0.901	0.934	3^{th}
	A1	0.691	0.758	0.855	0.901	0.934	3^{th}
	A2	0.684	0.750	0.847	0.898	0.931	2^{nd}
	A3	**0.636**	**0.701**	**0.819**	**0.799**	**0.839**	1^{st}
F_2	*A0*	0.773	0.876	0.917	0.962	0.998	3^{th}
	A1	0.773	0.876	0.917	0.962	0.998	3^{th}
	A2	0.763	0.835	0.918	0.959	0.995	2^{nd}
	A3	**0.738**	**0.764**	**0.876**	**0.860**	**0.897**	1^{st}
F_3	*A0*	0.603	0.653	0.750	0.806	0.840	3^{th}
	A1	0.603	0.653	0.750	0.806	0.840	3^{th}
	A2	0.593	0.652	0.741	0.797	0.827	2^{nd}
	A3	**0.534**	**0.593**	**0.677**	**0.673**	**0.708**	1^{st}

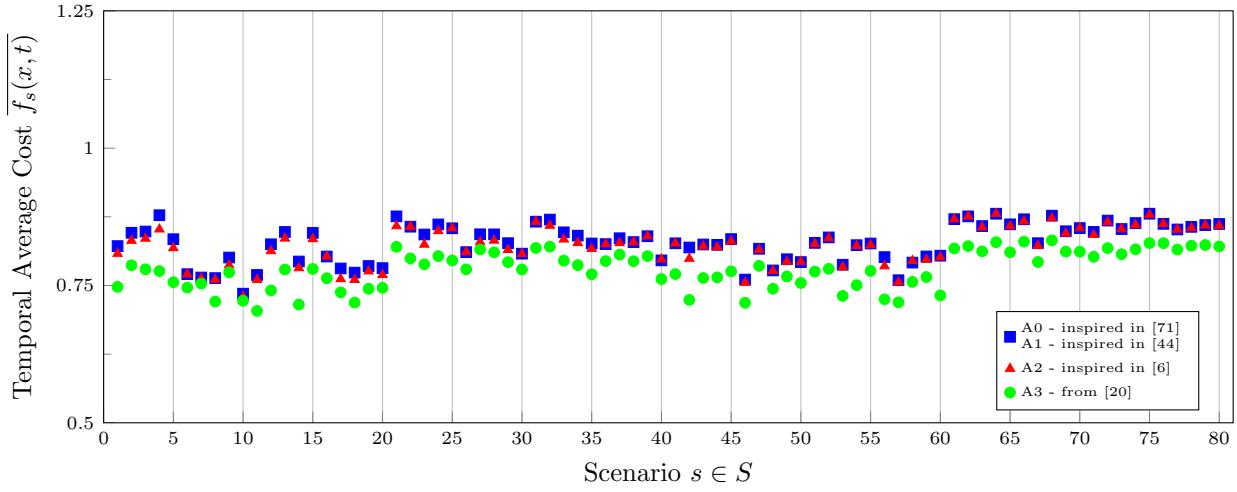

Figure 3: Temporal average cost from [20]: Average values of $\overline{f_s(x,t)}$ in DC_1 to DC_5 per each $s \in S$.

Based on the information presented in Table 5, the *Main Findings* (MFs) of the experimental evaluation performed in this section are summarized as follows:

Algorithm A3 that considered the proposed VMPr Triggering and VMPr Recovering methods outperformed all other evaluated algorithms in every experiment, taking into account the considered evaluation criteria.

In summary, *A3* obtained better results (minimum cost) for the evaluation criteria presented in Table 5.

When considering average objective function costs (F_1) as evaluation criterion, *A3* obtained between 3.4% and 12.4% better results than *A2*, as well as between 4.4% and 12.9% better results than *A0* and *A1*. Additionally, when considering maximum objective function costs (F_2) as evaluation criterion, the proposed *A3* obtained between 3.3% and 14.1% better results than *A2*, which performed as the second best algorithm in this case. When comparing to *A0* and *A1*, the proposed *A3* algorithm obtained between 4.7% and 15.4% better results. Finally, *A3* obtained between 9.9% and 11.4% better results than *A2* when considering minimum objective function costs (F_3) as evaluation criterion. *A3* algorithm also obtained between 10.1% and 14.7% better results than *A0* and *A1*.

To better understand the experimental evaluation summarized in Table 5, Figure 3 illustrates the temporal average cost of the single combined objective function for all scenarios $s \in S$, denoted as $\overline{f_s(x,t)}$ in Equation (12).

The proposed A3 outperformed other evaluated algorithms in the considered scenarios, when considering average values of the single combined objective function on each scenario $s \in S$.

As presented in Figure 3, *A3* outperformed the other algorithms in all of the considered scenarios. *A3* was the best algorithm in 100% of the 400 carefully designed and evaluated scenarios with different load of CPU resources.

Table 6: Summary of evaluation criteria in experimental results for evaluated algorithms.

Criterion	Algorithm	Datacenter					
		DC_1	DC_2	DC_3	DC_4	DC_5	Ranking
F_1	**A3**	**0.636**	**0.701**	**0.819**	**0.799**	**0.839**	1^{st}
	A4	0.794	0.932	0.986	1.003	1.019	2^{nd}

Summarizing, according to the performed experimental evaluation, the algorithm that considered the proposed prediction-based VMPr Triggering and update-based VMPr Recovering methods ($A3$) is the clear alternative for solving the uncertain MaVMP problem in a two-phase optimization scheme, considering results presented in this section.

6.5 Experimental Results: Comparison with OpenStack

Considering the promising results obtained in [20] and presented in Section 6.4, as additional contributions of this work, a brief initial comparison of the proposed $A3$ against an industry de-facto standard algorithm (i.e. inspired in OpenStack - $A4$) is presented in this section, considering the F_1 costs (see Equation (13)).

Table 6 presents values of the considered evaluation criteria, i.e. F_1 costs (see Equation (13)), summarizing results obtained in first performed simulations. The mentioned evaluation criterion is presented separately for each of the five considered IaaS cloud datacenter.

Based on the information presented in Table 6, the *Main Findings* (MFs) of the experimental evaluation performed in this section are summarized as follows:

Algorithm $A3$ that considered the proposed VMPr Triggering and VMPr Recovering methods outperformed other evaluated algorithm $A4$ in every experiment, taking into account the considered evaluation criterion.

When considering average objective function costs (F_1) as evaluation criterion, $A3$ obtained between 20% and 32% better results than $A4$. In summary, $A3$ obtained better results (minimum cost) for evaluation criterion presented in Table 6.

7 Conclusions and Future Directions

Based on 84 studied articles, the research summarized in this paper presented general taxonomies of the VMP problem from [8, 9, 10], considering possible environments where the VMP problem could be studied (Figure 2.1) as well as formulations and techniques for the resolution of the VMP problem (Table 1).

A formulation of a MaVMP for initial placement of VMs was also presented, considering the simultaneous optimization of five objective functions [15, 16, 18]). Adjustable constraints on upper and lower limits of each objective function are also recommended as a way to interactively control a potential explosion in the number of solutions. An interactive MA was additionally proposed to solve the proposed formulation, validating the formulation and proving that it is solvable.

Next, this work presented a first MaVMP with reconfiguration of VMs from [17], simultaneously optimizing five objective functions. An extended MA was proposed to solve this semi-dynamic MaVMP. Five selection strategies were evaluated to automate the process of selecting a solution from a Pareto set approximation P_{known} at each discrete instant t. Experimental results recommend the $S3$ (minimum distance to origin) strategy.

Additionally, a complex IaaS environment for VMP problems was proposed, considering service elasticity, including both vertical and horizontal scaling of cloud services, as well as overbooking of physical resources, including both server (CPU and RAM) and networking resources [20]). The proposed complex IaaS environment for VMP problems was studied in a two-phase optimization scheme, combining advantages of both online and offline VMP formulations, where novel prediction-based VMPr Triggering and update-based VMPr Recovering methods were proposed [20]). Experimental results suggested that the best algorithm for solving the proposed uncertain VMP problem is the one considering the proposed methods used by the $A9$ algorithm.

Several future works were also identified, including considering a dynamic set of PMs $H(t)$, as well as more sophisticated cloud federation approaches. Additionally, an experimental evaluation of alternative algorithms for both iVMP and VMPr phase is proposed as a future work, in order to explore performance issues with the proposed VMPr Triggering and VMPr Recovering methods. Novel VMPr methods could still be proposed to improve the considered two-phase optimization scheme. A more detailed experimental evaluation of different parameters of the proposed VMP formulation should also be considered, evaluating different protection factors λ_k, penalty factors ϕ_k or even different scalarization methods [75].

Additionally, the authors were working on implementing the evaluated algorithms in IaaS middlewares (e.g. OpenStack[1]) to evaluate the proposed methods in real-world cloud computing datacenters supporting real workloads of cloud applications. In this context, novel contributions were also included to compare proposed algorithms against simulated OpenStack inspired alternatives, presenting also promising results.

The following challenges are presented to guide further advance on this research:

- Considering that cloud-native applications are mainly using container-based deployment schemes, novel considerations for the VMP problem should be studied, taking into account particular architectures related to containers over VMs, such as the ones presented in [76].
- Answering research questions related to the possible application of machine learning techniques for characterising and also predicting the operation patterns of cloud computing datacenters is a relevant topic to investigate [77], mainly considering the dynamic nature of these mentioned patterns and limitations of machine learning techniques with this type of dynamic patterns.
- Serverless ecosystems [78] are emerging and represent a relevant topic that should be included in terms of its implications and granularity in VMP problems for cloud computing environments. How should be modelled and resolved serverless operations from the provider perspective?

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Generation computer systems*, vol. 25, no. 6, pp. 599–616, 2009, DOI: 10.1016/j.future.2008.12.001.

[2] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," *National Institute of Standards and Technology*, vol. 53, no. 6, p. 50, 2009.

[3] S. S. Manvi and G. K. Shyam, "Resource management for infrastructure as a service (IaaS) in cloud computing: A survey," *Journal of Network and Computer Applications*, vol. 41, pp. 424–440, 2014, DOI: 10.1016/j.jnca.2013.10.004.

[4] E. Elmroth, J. Tordsson, F. Hernández, A. Ali-Eldin, P. Svärd, M. Sedaghat, and W. Li, "Self-management challenges for multi-cloud architectures," in *Towards a Service-Based Internet*. Springer, 2011, pp. 38–49, DOI: 10.1007/978-3-642-24755-2-4.

[5] A. Anand, J. Lakshmi, and S. Nandy, "Virtual machine placement optimization supporting performance SLAs," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 1. IEEE, 2013, pp. 298–305, DOI: 10.1109/CloudCom.2013.46.

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

[7] A. Beloglazov and R. Buyya, "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*, vol. 24, no. 13, pp. 1397–1420, 2012, DOI: 10.1002/cpe.1867.

[8] F. López-Pires and B. Barán, "A virtual machine placement taxonomy," in *Cluster, Cloud and Grid Computing (CCGrid), 2015 15th IEEE/ACM International Symposium on*. IEEE Computer Society, May 2015, pp. 159–168, DOI: 10.1109/CCGrid.2015.15.

[9] ——, "Cloud computing resource allocation taxonomies," *International Journal of Cloud Computing*, vol. 6, no. 3, pp. 238–264, 2017, DOI: 10.1504/IJCC.2017.086712.

[10] B. Barán and F. López-Pires, "Resource allocation for cloud infrastructures: Taxonomies and research challenges," in *Research Advances in Cloud Computing*, S. Chaudhary, G. Somani, and R. Buyya, Eds. Springer, 2017, ch. 10, pp. 266–290, DOI: 10.1007/978-981-10-5026-8-11.

[11] J. Cheng, G. G. Yen, and G. Zhang, "A many-objective evolutionary algorithm based on directional diversity and favorable convergence," in *Systems, Man and Cybernetics (SMC), 2014 IEEE International Conference on*, Oct 2014, pp. 2415–2420, DOI: 10.1109/SMC.2014.6974288.

[1] http://www.openstack.org

[12] M. Farina and P. Amato, "On the optimal solution definition for many-criteria optimization problems," in *Proceedings of the NAFIPS-FLINT international conference*, 2002, pp. 233–238, DOI: 10.1109/NAFIPS.2002.1018061.

[13] K. Deb, A. Sinha, and S. Kukkonen, "Multi-objective test problems, linkages, and evolutionary methodologies," in *Proceedings of the 8th annual conference on Genetic and evolutionary computation*. ACM, 2006, pp. 1141–1148, DOI: 10.1145/1143997.1144179.

[14] M. Guzek, P. Bouvry, and E.-G. Talbi, "A survey of evolutionary computation for resource management of processing in cloud computing," *Computational Intelligence Magazine, IEEE*, vol. 10, no. 2, pp. 53–67, 2015, DOI: 10.1109/MCI.2015.2405351.

[15] F. López-Pires and B. Barán, "A many-objective optimization framework for virtualized datacenters," in *Proceedings of the 2015 5th International Conference on Cloud Computing and Service Science*, 2015, pp. 439–450, DOI: 10.5220/0005434604390450.

[16] ——, "Many-objective virtual machine placement," *Journal of Grid Computing*, vol. 15, no. 2, pp. 161–176, 2017, DOI: 10.1007/s10723-017-9399-x.

[17] D. Ihara, F. López-Pires, and B. Barán, "Many-objective virtual machine placement for dynamic environments," in *2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC)*. IEEE, 2015, pp. 75–79, DOI: 10.1109/UCC.2015.22.

[18] F. López-Pires and B. Barán, "Many-objective optimization for virtual machine placement in cloud computing," in *Research Advances in Cloud Computing*, S. Chaudhary, G. Somani, and R. Buyya, Eds. Springer, 2017, ch. 11, pp. 266–290, DOI: 10.1007/978-981-10-5026-8-12.

[19] F. López-Pires, B. Barán, A. Amarilla, L. Benítez, R. Ferreira, and S. Zalimben, "An experimental comparison of algorithms for virtual machine placement considering many objectives," in *9th Latin America Networking Conference (LANC)*, 2016, pp. 75–79, DOI: 10.1145/2998373.2998374.

[20] F. López-Pires, B. Barán, L. Benítez, S. Zalimben, and A. Amarilla, "Virtual machine placement for elastic infrastructures in overbooked cloud computing datacenters under uncertainty," *Future Generation Computer Systems*, vol. 79, no. 3, pp. 830–848, 2018, DOI: 10.1016/j.future.2017.09.021.

[21] L. Salimian and F. Safi, "Survey of energy efficient data centers in cloud computing," in *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 2013, pp. 369–374, DOI: 10.5555/2588611.2588641.

[22] M. Gahlawat and P. Sharma, "Survey of virtual machine placement in federated clouds," in *Advance Computing Conference (IACC), 2014 IEEE International*. IEEE, 2014, pp. 735–738, DOI: 10.1109/IAdCC.2014.6779415.

[23] K. Mills, J. Filliben, and C. Dabrowski, "Comparing VM-placement algorithms for on-demand clouds," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 91–98, DOI: 10.1109/CloudCom.2011.22.

[24] F. López-Pires and B. Barán, "Virtual machine placement literature review," *http://arxiv.org/abs/1506.01509*, 2015.

[25] F. López-Pires and B. Barán, "Multi-objective virtual machine placement with service level agreement: A memetic algorithm approach," in *Proceedings of the 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing*. IEEE Computer Society, 2013, pp. 203–210, DOI: 10.1109/UCC.2013.44.

[26] K. Sato, M. Samejima, and N. Komoda, "Dynamic optimization of virtual machine placement by resource usage prediction," in *Industrial Informatics (INDIN), 2013 11th IEEE International Conference on*. IEEE, 2013, pp. 86–91, DOI: 10.1109/INDIN.2013.6622863.

[27] K.-y. Chen, Y. Xu, K. Xi, and H. J. Chao, "Intelligent virtual machine placement for cost efficiency in geo-distributed cloud systems," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3498–3503, DOI: 10.1109/ICC.2013.6655092.

[28] E. Bin, O. Biran, O. Boni, E. Hadad, E. K. Kolodner, Y. Moatti, and D. H. Lorenz, "Guaranteeing high availability goals for virtual machine placement," in *Distributed Computing Systems (ICDCS), 2011 31st International Conference on*. IEEE, 2011, pp. 700–709, DOI: 10.1109/ICDCS.2011.72.

[29] C. Dupont, G. Giuliani, F. Hermenier, T. Schulze, and A. Somov, "An energy aware framework for virtual machine placement in cloud federated data centres," in *Future Energy Systems: Where Energy, Computing and Communication Meet (e-Energy), 2012 Third International Conference on.* IEEE, 2012, pp. 1–10, DOI: 10.1145/2208828.2208832.

[30] C. C. T. Mark, D. Niyato, and T. Chen-Khong, "Evolutionary optimal virtual machine placement and demand forecaster for cloud computing," in *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on.* IEEE, 2011, pp. 348–355, DOI: 10.1109/AINA.2011.50.

[31] H. Goudarzi and M. Pedram, "Energy-efficient virtual machine replication and placement in a cloud computing system," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on.* IEEE, 2012, pp. 750–757, DOI: 10.1109/CLOUD.2012.107.

[32] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in *Grid and Cooperative Computing (GCC), 2010 9th International Conference on.* IEEE, 2010, pp. 87–92, DOI: 10.1109/GCC.2010.29.

[33] H. T. Dang and F. Hermenier, "Higher SLA satisfaction in datacenters with continuous VM placement constraints," in *Proceedings of the 9th Workshop on Hot Topics in Dependable Systems.* ACM, 2013, p. 1, DOI: 10.1145/2524224.2524226.

[34] W. Li, J. Tordsson, and E. Elmroth, "Virtual machine placement for predictable and time-constrained peak loads," in *Economics of Grids, Clouds, Systems, and Services.* Springer, 2012, pp. 120–134, DOI: 10.1007/978-3-642-28675-9-9.

[35] M. Sun, W. Gu, X. Zhang, H. Shi, and W. Zhang, "A matrix transformation algorithm for virtual machine placement in cloud," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2013 12th IEEE International Conference on.* IEEE, 2013, pp. 1778–1783, DOI: 10.1109/TrustCom.2013.221.

[36] F. Song, D. Huang, H. Zhou, H. Zhang, and I. You, "An optimization-based scheme for efficient virtual machine placement," *International Journal of Parallel Programming*, vol. 42, no. 5, pp. 853–872, 2014, DOI: 10.1007/s10766-013-0274-5.

[37] D. S. Dias and L. H. M. Costa, "Online traffic-aware virtual machine placement in data center networks," in *Global Information Infrastructure and Networking Symposium (GIIS), 2012.* IEEE, 2012, pp. 1–8, DOI: 10.1109/GIIS.2012.6466665.

[38] W. Shi and B. Hong, "Towards profitable virtual machine placement in the data center," in *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on.* IEEE, 2011, pp. 138–145, DOI: 10.1109/UCC.2011.28.

[39] A. Gupta, L. V. Kalé, D. Milojicic, P. Faraboschi, and S. M. Balle, "HPC-Aware vm placement in infrastructure clouds," in *Cloud Engineering (IC2E), 2013 IEEE International Conference on.* IEEE, 2013, pp. 11–20, DOI: 10.1109/IC2E.2013.38.

[40] J. Dong, H. Wang, X. Jin, Y. Li, P. Zhang, and S. Cheng, "Virtual machine placement for improving energy efficiency and network performance in IaaS cloud," in *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference on.* IEEE, 2013, pp. 238–243, DOI: 10.1109/ICDCSW.2013.48.

[41] H.-J. Hong, D.-Y. Chen, C.-Y. Huang, K.-T. Chen, and C.-H. Hsu, "Qoe-aware virtual machine placement for cloud games," in *Network and Systems Support for Games (NetGames), 2013 12th Annual Workshop on.* IEEE, 2013, pp. 1–2, DOI: 10.1109/NetGames.2013.6820610.

[42] A. Dalvandi, M. Gurusamy, and K. C. Chua, "Time-aware vm-placement and routing with bandwidth guarantees in green cloud data centers," in *Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on*, vol. 1. IEEE, 2013, pp. 212–217, DOI: 10.1109/CloudCom.2013.36.

[43] W. Wang, H. Chen, and X. Chen, "An availability-aware virtual machine placement approach for dynamic scaling of cloud applications," in *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on.* IEEE, 2012, pp. 509–516, DOI: 10.1109/UIC-ATC.2012.31.

[44] N. M. Calcavecchia, O. Biran, E. Hadad, and Y. Moatti, "Vm placement strategies for cloud scenarios," in *Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on*. IEEE, 2012, pp. 852–859, DOI: 10.1109/CLOUD.2012.113.

[45] Z. Cao and S. Dong, "An energy-aware heuristic framework for virtual machine consolidation in cloud computing," *The Journal of Supercomputing*, pp. 1–23, 2014, DOI: 10.1007/s11227-014-1172-3.

[46] M. Tang and S. Pan, "A hybrid genetic algorithm for the energy-efficient virtual machine placement problem in data centers," *Neural Processing Letters*, pp. 1–11, 2014, DOI: 10.1007/s11063-014-9339-8.

[47] K. Tsakalozos, M. Roussopoulos, and A. Delis, "Vm placement in non-homogeneous iaas-clouds," in *Service-Oriented Computing*. Springer, 2011, pp. 172–187, DOI: 10.1007/978-3-642-25535-9-12.

[48] J. Xu and J. A. Fortes, "Multi-objective virtual machine placement in virtualized data center environments," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on & Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*. IEEE, 2010, pp. 179–188, DOI: 10.1109/GreenCom-CPSCom.2010.137.

[49] S. Shigeta, H. Yamashima, T. Doi, T. Kawai, and K. Fukui, "Design and implementation of a multi-objective optimization mechanism for virtual machine placement in cloud computing data center," in *Cloud Computing*. Springer, 2013, pp. 21–31, DOI: 10.1007/978-3-319-03874-2-3.

[50] A. C. Adamuthe, R. M. Pandharpatte, and G. T. Thampi, "Multiobjective virtual machine placement in cloud environment," in *Cloud & Ubiquitous Computing & Emerging Technologies (CUBE), 2013 International Conference on*. IEEE, 2013, pp. 8–13, DOI: 10.1109/CUBE.2013.12.

[51] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing," *Journal of Computer and System Sciences*, vol. 79, no. 8, pp. 1230–1242, 2013, DOI: 10.1016/j.jcss.2013.02.004.

[52] F. López-Pires, E. Melgarejo, and B. Barán, "Virtual machine placement. A multi-objective approach," in *Computing Conference (CLEI), 2013 XXXIX Latin American*. IEEE, 2013, pp. 1–8, DOI: 10.1109/CLEI.2013.6670671.

[53] J.-J. Wu, P. Liu, and J.-S. Yang, "Workload characteristics-aware virtual machine consolidation algorithms," in *Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE Computer Society, 2012, pp. 42–49, DOI: 10.1109/CloudCom.2012.6427540.

[54] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "Vmplanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Computer Networks*, vol. 57, no. 1, pp. 179–196, 2013, DOI: 10.1016/j.comnet.2012.09.008.

[55] C. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary algorithms for solving multi-objective problems*. Springer, 2007, DOI: 10.1007/978-0-387-36797-2.

[56] M. Báez, D. Zárate, and B. Barán, "Adaptive memetic algorithms for multi-objective optimization," in *Computing Conference (CLEI), 2007 XXXIII Latin American*, vol. 2007, 2007.

[57] C. von Lücken, B. Barán, and C. Brizuela, "A survey on multi-objective evolutionary algorithms for many-objective problems," *Computational Optimization and Applications*, pp. 1–50, 2014, DOI: 10.1007/s10589-014-9644-1.

[58] L. Shi, B. Butler, D. Botvich, and B. Jennings, "Provisioning of requests for virtual machine sets with placement constraints in iaas clouds," in *Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on*. IEEE, 2013, pp. 499–505.

[59] W. Li, J. Tordsson, and E. Elmroth, "Modeling for dynamic cloud scheduling via migration of virtual machines," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 163–171.

[60] J. Ortigoza, F. López-Pires, and B. Barán, "A taxonomy on dynamic environments for provider-oriented virtual machine placement," in *2016 IEEE International Conference on Cloud Engineering (IC2E)*, April 2016, pp. 214–215.

[61] B. Speitkamp and M. Bichler, "A mathematical programming approach for server consolidation problems in virtualized data centers," *Services Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 266–278, 2010, DOI: 10.1109/TSC.2010.25.

[62] W. Yue and Q. Chen, "Dynamic placement of virtual machines with both deterministic and stochastic demands for green cloud computing," *Mathematical Problems in Engineering*, vol. 2014, 2014, DOI: 10.1155/2014/613719.

[63] E. Feller, C. Morin, and A. Esnault, "A case for fully decentralized dynamic vm consolidation in clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*. IEEE, 2012, pp. 26–33, DOI: 10.1109/CloudCom.2012.6427585.

[64] X.-F. Liu, Z.-H. Zhan, K.-J. Du, and W.-N. Chen, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach," in *Proceedings of the 2014 conference on Genetic and evolutionary computation*. ACM, 2014, pp. 41–48, DOI: 10.1145/2576768.2598265.

[65] F. Farahnakin, R. Bahsoon, P. Liljeberg, and T. Pahikkala, "Self-adaptive resource management system in iaas clouds," in *9th International Conference on Cloud Computing (IEEE CLOUD)*, R. Bahsoon, P. Liljeberg, and T. Pahikkala, Eds. IEEE, 2016, pp. 553–560, DOI: 10.1109/CLOUD.2016.0079.

[66] Q. Zheng, R. Li, X. Li, N. Shah, J. Zhang, F. Tian, K.-M. Chao, and J. Li, "Virtual machine consolidated placement based on multi-objective biogeography-based optimization," *Future Generation Computer Systems*, vol. 54, pp. 95–122, 2016, DOI: 10.1016/j.future.2015.02.010.

[67] P. Sv, W. Li, E. Wadbro, J. Tordsson, E. Elmroth *et al.*, "Continuous datacenter consolidation," in *2015 IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*. IEEE, 2015, pp. 387–396, DOI: 10.1109/CloudCom.2015.11.

[68] J. Shi, F. Dong, J. Zhang, J. Luo, and D. Ding, "Two-phase online virtual machine placement in heterogeneous cloud data center," in *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1369–1374, DOI: 10.1109/SMC.2015.243.

[69] M. Tighe and M. Bauer, "Integrating cloud application autoscaling with dynamic vm allocation," in *2014 IEEE Network Operations and Management Symposium (NOMS)*. IEEE, 2014, pp. 1–9, DOI: 10.1109/NOMS.2014.6838239.

[70] M. A. Aloulou and F. Della Croce, "Complexity of single machine scheduling problems under scenario-based uncertainty," *Operations Research Letters*, vol. 36, no. 3, pp. 338–342, 2008, DOI: 10.1016/j.orl.2007.11.005.

[71] S. Fang, R. Kanagavelu, B.-S. Lee, C. H. Foh, and K. M. M. Aung, "Power-efficient virtual machine placement and migration in data centers," in *Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing*. IEEE, 2013, pp. 1408–1413, DOI: 10.1109/GreenCom-iThings-CPSCom.2013.246.

[72] D. OpenStack, "Scheduling," https://docs.openstack.org/mitaka/config-reference/compute/scheduler.html, 2018, [Online; accessed 14-June-2018].

[73] J. Huang, C. Li, and J. Yu, "Resource prediction based on double exponential smoothing in cloud computing," in *2012 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet)*, April 2012, pp. 2056–2060, DOI: 10.1109/CECNet.2012.6201461.

[74] D. P. Pinto-Roa, C. A. Brizuela, and B. Barán, "Multi-objective routing and wavelength converter allocation under uncertain traffic," *Optical Switching and Networking*, vol. 16, pp. 1–20, 2015, DOI: 10.1016/j.osn.2014.10.001.

[75] A. Amarilla, S. Zalimben, L. Benítez, F. López-Pires, and B. Barán, "Evaluating a two-phase virtual machine placement optimization scheme for cloud computing datacenters," in *2017 Metaheuristics International Conference (MIC)*, 2017, pp. 99–108, DOI: 10.1109/CLEI.2017.8226393.

[76] R. Buyya, M. A. Rodriguez, A. N. Toosi, and J. Park, "Cost-efficient orchestration of containers in clouds: a vision, architectural elements, and future directions," in *Journal of Physics: Conference Series*, vol. 1108, no. 1. IOP Publishing, 2018, p. 012001.

[77] F. López-Pires and B. Barán, "Machine learning opportunities in cloud computing data center management for 5g services," in *2018 ITU Kaleidoscope: Machine Learning for a 5G Future (ITU K)*. IEEE, 2018, pp. 1–6, DOI: 10.23919/ITU-WT.2018.8597920.

[78] Y. Bogado-Sarubbi, W. Benitez-Davalos, J. Spillner, and F. Lopez-Pires, "Towards sustainable ecosystems for cloud functions," in *ESSCA, Zurich, 21 December 2018*. CEUR-WS, 2019, pp. 18–24, DOI: 10.21256/zhaw-3270.