Power-efficient Assignment of Virtual Machines to Physical Machines[☆]

Jordi Arjona Aroca[a,b], Antonio Fernández Anta[a], Miguel A. Mosteiro[c], Christopher Thraves[d], Lin Wang[e,f]

[a]*Institute IMDEA Networks, Madrid, Spain*
[b]*Universidad Carlos III de Madrid, Madrid, Spain*
[c]*Department of Computer Science, Kean University, Union, NJ, USA*
[d]*CNRS-LAAS and Univ. Toulouse - LAAS, Tolouse, France*
[e]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China*
[f]*University of Chinese Academy of Sciences, Beijing, China*

Abstract

Motivated by current trends in cloud computing, we study a version of the generalized assignment problem where a set of virtual processors has to be implemented by a set of *identical* processors. For literature consistency, we say that a set of virtual machines (VMs) is assigned to a set of physical machines (PMs). The optimization criterion is to minimize the power consumed by all the PMs. We term the problem Virtual Machine Assignment (VMA). Crucial differences with previous work include a variable number of PMs, that each VM must be assigned to exactly one PM (i.e., VMs cannot be implemented fractionally), and a minimum power consumption for each active PM. Such infrastructure may be strictly constrained in the number of PMs or in the PMs' capacity, depending on how costly (in terms of power consumption) it is to add a new PM to the system or to heavily load some of the existing PMs. Low usage or ample budget yields models where PM capacity and/or the number of PMs may be assumed unbounded for all practical purposes. We study four VMA problems depending on whether the capacity or the number of PMs is bounded or not. Specifically, we study hardness and online competitiveness for a variety of cases. To the best of our knowledge, this is the first comprehensive study of the VMA problem for this cost function.

Keywords: cloud computing, generalized assignment, scheduling, load balancing

1. Introduction

The current pace of technology developments, and the continuous change in business require-ments, may rapidly yield a given proprietary computational platform obsolete, oversized, or in-sufficient. Thus, outsourcing has recently become a popular approach to obtain computational services without incurring in amortization costs. Furthermore, in order to attain flexibility, such service is usually virtualized, so that the user may tune the computational platform to its particular needs. Users of such service need not to be aware of the particular implementation, they only need to specify the virtual machine they want to use. This conceptual approach to outsourced computing has been termed *cloud computing*, in reference to the cloud symbol used as an abstraction of a com-plex infrastructure in system diagrams. Current examples of cloud computing providers include Amazon Web Services [EC2, 2012], Rackspace [Rackspace, 2012], and Citrix [Citrix, 2012].

Depending on what the specific service provided is, the cloud computing model comes in different flavors, such as *infrastructure as a service*, *platform as a service*, *storage as a service*, etc. In each of these models, the user may choose specific parameters of the computational resources provided. For instance, processing power, memory size, communication bandwidth, etc. Thus, in a cloud-computing service platform, various ***virtual machines (VM)*** with user-defined specifications must be implemented by, or ***assigned to***[1], various ***physical machines (PM)***[2]. Furthermore, such a platform must be scalable, allowing to add more PMs, should the business growth require such expansion. In this work, we call this problem the ***Virtual Machine Assignment (VMA)*** problem.

The optimization criteria for VMA depends on what the particular objective function sought is. From the previous discussion, it can be seen that, underlying VMA, there is some form of bin-packing problem. However, in VMA the number of PMs (i.e., bins for bin packing) may be in-creased if needed. Since CPU is generally the dominant power consumer in a server [Arjona Aroca et al., 2014a], VMA is usually carried out according to CPU workloads. With only the static power consumption of servers considered, previous work related to VMA has focused on minimizing the number of active PMs (cf. [Bellur et al., 2010] and the references therein) in order to minimize

---

[1] The cloud-computing literature use instead the term *placement*. We choose here the term assignment for consis-tency with the literature on general assignment problems.

[2] We choose the notation VM and PM for simplicity and consistency, but notice that our study applies to any computational resource assignment problem, as long as the minimization function is the one modeled here.

the total static energy consumption. This is commonly known as VM consolidation [Nathuji and Schwan, 2007; Kusic et al., 2009]. However, despite the static power, the dynamic power consumption of a server, which has been shown to be superlinear on the **load** of a given computational resource [Bansal et al., 2009; Gupta et al., 2012], is also significant and cannot be ignored. Since the definition of load is not precise, we borrow the definition in [Arjona Aroca et al., 2014a] and define the load of a server as the amount of active cycles per second a task requires, an absolute metric independent of the operating frequency or the number of cores of a PM. The superlinearity property of the dynamic power consumption is also confirmed by the results in [Arjona Aroca et al., 2014a]. As a result, when taking into account both parts of power consumption, the use of extra PMs may be more efficient energy-wise than a minimum number of heavily-loaded PMs. This inconsistency with the literature in VM consolidation has been supported by the results presented in [Arjona Aroca et al., 2014a] and, hence, we claim that the way consolidation has been traditionally performed has to be reconsidered. In this work, we combine both power-consumption factors and explore the most energy-efficient way for VMA. That is, for some parameters $\alpha > 1$ and $b > 0$, we seek to minimize the sum of the α powers of the PMs loads *plus* the fixed cost b of using each PM.

Physical resources are physically constrained. A PMs infrastructure may be strictly constrained in the number of PMs or in the PMs CPU capacity. However, if usage patterns indicate that the PMs will always be loaded well below their capacity, it may be assumed that the capacity is unlimited. Likewise, if the power budget is very big, the number of PMs may be assumed unconstrained for all practical purposes. These cases yield 4 VMA subproblems, depending on whether the capacity and the number of PMs is limited or not. We introduce these parameters denoting the problem as **(C,m)-VMA**, where C is the PM CPU capacity, m is the maximum number of PMs, and each of these parameters is replaced by a dot if unbounded.

In this work, we study the hardness and online competitiveness of the VMA problem. Specifically, we show that VMA is NP-hard *in the strong sense* (in particular, we observe that (C, m)-VMA is strongly NP-complete). Thus, VMA problems do not have a fully polynomial time approximation scheme (FPTAS). Nevertheless, using previous results derived for more general objective functions, we notice that $(\cdot, m)$- and $(\cdot, \cdot)$-VMA have a polynomial time approximation scheme (PTAS). We also show various lower and upper bounds on the offline approximation and

the online competitiveness of VMA. Rather than attempting to obtain tight bounds for particular instances of the parameters of the problem (C, m, α, b) we focus on obtaining *general bounds*, whose parameters can be instantiated for the specific application. The bounds obtained show interesting trade-offs between the PM capacity and the fixed cost of adding a new PM to the system. To the best of our knowledge, this is the first VMA study that is focused on power consumption.

Roadmap. The paper is organized as follows. In what remains of this section, we define formally the $(\cdot, \cdot)$-VMA problem, we overview the related work, and we describe our results in detail. Section 2 includes some preliminary results that will be used throughout the paper. The offline and online analyses are included in Section 3 and 4 respectively. Section 5 discusses some practical issues and provides some useful insights regarding real implementation.

1.1. Problem Definition

We describe the $(\cdot, \cdot)$-VMA problem now. Given a set $S = \{s_1, \ldots, s_m\}$ of $m > 1$ identical physical machines (PMs) of capacity C; rational numbers μ, α and b, where $\mu > 0$, $\alpha > 1$ and $b > 0$; a set $D = \{d_1, \ldots, d_n\}$ of n virtual machines and a function $\ell : D \to \mathbb{R}$ that gives the CPU load each virtual machine incurs[3], we aim to obtain a partition $\pi = \{A_1, \ldots, A_m\}$ of D, such that $\ell(A_i) \leq C$, for all i. Our objective will be then minimizing the power consumption given by the function

$$P(\pi) = \sum_{i \in [1,m] : A_i \neq \emptyset} \left(\mu \Big(\sum_{d_j \in A_i} \ell(d_j) \Big)^{\alpha} + b \right). \tag{1}$$

Let us define the function $f(\cdot)$, such that $f(x) = 0$ if $x = 0$ and $f(x) = \mu x^{\alpha} + b$ otherwise. Then, the objective function is to minimize $P(\pi) = \sum_{i=1}^{m} f(\ell(A_i))$. The parameter μ is used for consistency with the literature. For clarity we will consider $\mu = 1$ in the rest of the paper. All the results presented apply for other values of μ.

We also study several special cases of the VMA problem, namely (C, m)-VMA, $(C, \cdot)$-VMA, $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA. (C, m)-VMA refers to the case where both the number of available PMs and its capacity are fixed. $(\cdot, \cdot)$-VMA, where $(\cdot)$ denotes unboundedness, refers to the case where both the number of available PMs and its capacity are unbounded (i.e., C is larger than the

---

[3]For convenience, we overload the function $\ell(\cdot)$ to be applied over sets of virtual machines, so that for any set $A \subseteq D, \ell(A) = \sum_{d_j \in A} \ell(d_j)$.

total load of the VMs that can ever be in the system at any time, or m is larger than the number of VMs that can ever be in the system at any time). $(C, \cdot)$-VMA and $(\cdot, m)$-VMA are the cases where the number of available PMs and their capacity is unbounded, respectively.

1.2. Related Work

To the best of our knowledge, previous work on VMA has been only experimental [Chen et al., 2011; Liu et al., 2012; Mills et al., 2011a; Van den Bossche et al., 2011; Beloglazov et al., 2012; Espadas et al., 2013] or has focused on different cost functions [Cody and Jr., 1976; Alon et al., 1997; Bellur et al., 2010; Chandra and Wong, 1975]. First, we provide an overview of previous theoretical work for related assignment problems (storage allocation, scheduling, network design, etc.). The cost functions considered in that work resemble or generalize the power cost function under consideration here. Secondly, we overview related experimental work.

Chandra and Wong [Chandra and Wong, 1975], and Cody and Coffman [Cody and Jr., 1976] study a problem for storage allocation that is a variant of $(\cdot, m)$-VMA with $b = 0$ and $\alpha = 2$. Hence, this problem tries to minimize the sum of the squares of the machine-load vector for a fixed number of machines. They study the offline version of the problem and provide algorithms with constant approximation ratio. A significant leap was taken by Alon et al. [Alon et al., 1997], since they present a PTAS for the problem of minimizing the L_p norm of the load vector, for any $p \geq 1$. This problem has the previous one as special case, and is also a variant of the $(\cdot, m)$-VMA problem when $p = \alpha$ and $b = 0$. Similarly, Alon et al [Alon et al., 1998] extended this work for a more general set of functions, that include $f(\cdot)$ as defined above. Hence, their results can be directly applied in the $(\cdot, m)$-VMA problem. Later, Epstein et al [Epstein and Sgall, 2004] extended [Alon et al., 1998] further for the uniformly related machines case. We will use these results in Section 3 in the analysis of the offline case of $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA.

Bansal, Chan, and Pruhs minimize arbitrary power functions for speed scaling in job scheduling [Bansal et al., 2009]. The problem is to schedule the execution of n computational jobs on a *single* processor, whose speed may vary within a countable collection of intervals. Each job has a release time, a processing work to be done, a weight characterizing its importance, and its execution can be suspended and restarted later without penalty. A scheduler algorithm must specify, for each time, a job to execute and a speed for the processor. The goal is to minimize the weighted

5

sum of the flow times over all jobs plus the energy consumption, where the flow time of a job is the time elapsed from release to completion and the energy consumption is given by s^α where s is the processor speed and $\alpha > 1$ is some constant. For the online algorithm *shortest remaining processing time first*, the authors prove a $(3 + \epsilon)$ competitive ratio for the objective of total weighted flow plus energy. Whereas for the online algorithm *highest density first (HDF)*, where the density of a job is its weight-to-work ratio, they prove a $(2 + \epsilon)$ competitive ratio for the objective of fractional weighted flow plus energy.

Recently, Im, Moseley, and Pruhs studied online scheduling for general cost functions of the flow time, with the only restriction that such function is non-decreasing [Im et al., 2012]. In their model, a collection of jobs, each characterized by a release time, a processing work, and a weight, must be processed by a *single* server whose speed is variable. A job can be suspended and restarted later without penalty. The authors show that HDF is $(2 + \epsilon)$-speed $O(1)$-competitive against the optimal algorithm on a unit speed-processor, for all non-decreasing cost functions of the flow time. Furthermore, they also show that this ratio cannot be improved significantly proving impossibility results if the cost function is not uniform among jobs or the speed cannot be significantly increased.

A generalization of the above problem is studied by Gupta, Krishnaswamy, and Pruhs in [Gupta et al., 2012]. The question addressed is how to assign jobs, *possibly fractionally*, to unrelated parallel machines in an online fashion in order to minimize the sum of the α-powers of the machine loads plus the assignment costs. Upon arrival of a job, the algorithm learns the increase on the load and the cost of assigning a unit of such job to a machine. Jobs cannot be suspended and/or reassigned. The authors model a greedy algorithm that assigns a job so that the cost is minimized as solving a mathematical program with constraints arriving online. They show a competitive ratio of α^α with respect to the solution of the dual program which is a lower bound for the optimal. They also show how to adapt the algorithm to integral assignments with a $O(\alpha)^\alpha$ competitive ratio, which applies directly to our $(\cdot, m)$-VMA problem. References to previous work on the particular case of minimizing energy with deadlines can be found in this paper.

Similar cost functions have been considered for the minimum cost network-design problem. In this problem, packets have to be routed through a (possibly multihop) network of speed scalable routers. There is a cost associated to assigning a packet to a link and to the speed or load of the router. The goal is to route all packets minimizing the aggregated cost. In [Andrews et al., 2012]

and [Andrews et al., 2010] the authors show offline algorithms for this problem with undirected graph and homogeneous link cost functions that achieve polynomial and poly-logarithmic approximation, respectively. The cost function is the α-th power of the link load plus a link assignment cost, for any constant $\alpha > 1$. The same problem and cost function is studied in [Gupta et al., 2012]. Bansal *et al.* [Bansal et al., 2012] study a minimum-cost virtual circuit multicast routing problem with speed scalable links. They give a polynomial-time $O(\alpha)$-approximation offline algorithm and a polylog-competitive online algorithm, both for the case with homogeneous power functions. They also show that the problem is APX-hard in the case with heterogeneous power functions and there is no polylog-approximation when the graph is directed. Recently, Antoniadis *et al.* [Antoniadis et al., 2014] improved the results by providing a simple combinatorial algorithm that is $O(\log^{\alpha} n)$-approximate, from which we can construct an $\widetilde{O}(\log^{3\alpha+1} n)$-competitive online algorithm. The $(\cdot, m)$-VMA problem can be seen as a especial case of the problem considered in these papers in which there are only two nodes, source and destination, and m parallel links connecting them.

To the best of our knowledge, the problem of minimizing the power consumption (given in Eq.1) with capacity constraints (i.e., the (C, m)-VMA and $(C, \cdot)$-VMA problems) has received very limited attention, in the realm of both VMA and network design, although the approaches in [Andrews et al., 2010] and [Bansal et al., 2012] are related to or based on the solutions for the capacitated network-design problem [Chakrabarty et al., 2011].

The experimental work related to VMA is vast and its detailed overview is out of the scope of this paper. Some of this work does not minimize energy [Cardosa et al., 2011; Machida et al., 2010; Mishra and Sahoo, 2011] or it applies to a model different than ours (VM migration [Nguyen Van et al., 2009; Srikantaiah et al., 2008; Beloglazov et al., 2012], knowledge of future load [Mark et al., 2011; Srikantaiah et al., 2008], feasibility of allocation [Bellur et al., 2010], multilevel architecture [Mills et al., 2011a; Nguyen Van et al., 2009; Jansen and Brenner, 2011; Espadas et al., 2013], interconnected VMs [Botero et al., 2012], etc.). On the other hand, some of the experimental work where minimization of energy is evaluated focus on a more restrictive cost function [Viswanathan et al., 2011; Jansen and Brenner, 2011; Xu and Fortes, 2011].

In [Srikantaiah et al., 2008], the authors focus on an energy-efficient VM placement problem with two requirements: CPU and disk. These requirements are assumed to change dynamically

and the goal is to consolidate loads among servers, possibly using migration at no cost. In our model VMs assignment is based on a CPU requirement that does not change and migration is not allowed. Should any other resource be the dominating energy cost, the same results apply for that requirement. Also, if loads change and migration is free, an offline algorithm can be used each time that a load changes or a new VM arrives. In [Srikantaiah et al., 2008] it is shown experimentally that energy-efficient VMA does not merely reduce to a packing problem. That is, to minimize the number of PMs used even if their load is close to their maximum capacity. For our model, we show here that the optimal load of a given server is a function only of the fixed cost of being active (b) and the exponential rate of power increase on the load (α). That is, the optimal load is not related to the maximum capacity of a PM.

1.3. Overview of Results

In this work, we study offline and online versions of the four versions of the VMA problem. For the offline problems, the first fact we observe is that there is a hard decision version of (C, m)-VMA: Is there a feasible partition π of the set D of VMs? By reduction from the 3-Partition problem, it can be shown that this decision problem is strongly NP-complete.

We then show that the $(\cdot, \cdot)$-VMA, $(C, \cdot)$-VMA, and $(\cdot, m)$-VMA problems are NP-hard in the strong sense, even if α is constant. This result implies that these problems do not have FPTAS, even if α is constant. However, we show that the $(\cdot, \cdot)$-VMA and $(\cdot, m)$-VMA problems have PTAS, while the $(C, \cdot)$-VMA problem can not be approximated beyond a ratio of $\frac{3}{2} \cdot \frac{\alpha - 1 + (\frac{2}{3})^\alpha}{\alpha}$ (unless P = NP). On the positive side, we show how to use an existing Asymptotic PTAS [Fernandez de la Vega and Lueker, 1981] to obtain algorithms that approximate the optimal solution of $(C, \cdot)$-VMA. (See Table 1.)

Then we move on to online VMA algorithms. We show various upper and lower bounds on the competitive ratio of the four versions of the problem. (See Table 1.) Observe that the results are often different depending on whether x^* is smaller than C or not. In fact, when $x^* < C$, there is a lower bound of $\frac{(3/2)2^\alpha - 1}{2^\alpha - 1}$ that applies to all versions of the problem. The bounds are given as a function of the input parameters of the problem, in order to allow for tighter expressions. To provide intuition on how tight the bounds are, we instantiate them for a realistic [4] value of $\alpha = 3$,

---

[4]The values for α in the servers studied in [Arjona Aroca et al., 2014a] (denoted as Erdos and Nemesis) are close

VMA subprob.	$x^* < C$		$x^* \geq C$	
$(C,\cdot)$ offline	$\rho \geq \frac{3}{2}\frac{\alpha-1+(2/3)^\alpha}{\alpha}$	$\rho \geq \frac{11}{9}$	$\rho \geq \frac{3}{2}\frac{\alpha-1+(2/3)^\alpha}{\alpha}$	$\rho \geq \frac{11}{9}$
	$\rho < \frac{\overline{m}}{m^*}\left(1+\epsilon+\frac{1}{\alpha-1}+\frac{1}{\overline{m}}\right)$	$\rho < \frac{\overline{m}}{m^*}\left(\frac{3}{2}+\epsilon+\frac{1}{\overline{m}}\right)$	$\rho < 1+\epsilon+\frac{C^\alpha}{b}+\frac{1}{\overline{m}}$	$\rho < \frac{3}{2}+\epsilon+\frac{1}{\overline{m}}$
$(C,\cdot)$ online	$\rho \geq \frac{(3/2)2^\alpha-1}{2^\alpha-1}$	$\rho \geq \frac{11}{7}$	$\rho \geq \frac{C^\alpha+2b}{b+\max\{C^\alpha,2(C/2)^\alpha+b\}}$	$\rho \geq \frac{20}{17}$
	$\rho = 1$ if $D_s = \emptyset$, else $\rho \leq \left(1-\frac{1}{\alpha}\left(1-\frac{1}{2^\alpha}\right)\right)\left(2+\frac{x^*}{\ell(D_s)}\right)$	$\rho \leq \frac{17}{12}\left(1+\frac{1}{2\ell(D_s)}\right)$	$\rho \leq \frac{2b}{C}\left(1+\frac{1}{(\alpha-1)2^\alpha}\right)\left(2+\frac{C}{\ell(D)}\right)$	$\rho \leq \frac{17}{2}\left(1+\frac{1}{2\ell(D)}\right)$
(C,m) online	$\rho \geq \frac{(3/2)2^\alpha-1}{2^\alpha-1}$	$\rho \geq \frac{11}{7}$	$\rho \geq \frac{C^\alpha+2b}{b+\max\{C^\alpha,2(C/2)^\alpha+b\}}$	$\rho \geq \frac{20}{17}$
$(\cdot,\cdot)$ online	$\rho \geq \frac{(3/2)2^\alpha-1}{2^\alpha-1}$	$\rho \geq \frac{11}{7}$	not applicable	
	$\rho = 1$ if $D_s = \emptyset$, else $\rho \leq \left(1-\frac{1}{\alpha}\left(1-\frac{1}{2^\alpha}\right)\right)\left(2+\frac{x^*}{\ell(D_s)}\right)$	$\rho \leq \frac{17}{12}\left(1+\frac{1}{2\ell(D_s)}\right)$		
$(\cdot,m)$ online	$\rho \geq \max\{\frac{(3/2)2^\alpha-1}{2^\alpha-1}, \frac{3^\alpha}{2^{\alpha+2}+\epsilon}\}$	$\rho \geq \frac{11}{7}$	not applicable	
	$\rho \leq O(\alpha)^\alpha$ In [Gupta et al., 2012]			
$(\cdot,2)$ online	$\rho \geq \max\{\frac{3^\alpha}{2^{\alpha+1}}, \frac{(3/2)2^\alpha-1}{2^\alpha-1}, \frac{3^\alpha}{2^{\alpha+2}+\epsilon}\}$	$\rho \geq \frac{11}{7}$	not applicable	
	$\rho = 1$ if $\ell(D) \leq \sqrt[\alpha]{b/(2^\alpha-2)}$, else $\rho \leq \max\{2,\left(\frac{3}{2}\right)^{\alpha-1}\}$	$\rho \leq \frac{9}{4}$		

Table 1: Summary of bounds on the approximation/competitive ratio ρ. All lower bounds are existential. The number of PMs in an optimal $(C,\cdot)$-VMA solution is denoted as m^*. The number of PMs in an optimal Bin Packing solution is denoted as $\overline{m}$. The load that minimizes the ratio power consumption against load is denoted as x^*. The subset of VMs with load smaller than x^* is denoted as D_s. Shaded cells correspond to $\alpha = 3$, $b = 2$, and $C = 2$ on the left and $C = 1$ on the right.

and normalized values of $b = 2$ and $C \in \{1, 2\}$. The resulting bounds are shown in Table 1 in shaded cells. As can be observed, the resulting upper and lower bounds are not very far in general.

2. General Properties of the Model

The following claims will be used in the analysis. We call **_power rate_** the power consumed per unit of load in a PM. Let x be the load of a PM. Then, its power rate is computed as $f(x)/x$. The load at which the power rate is minimized, denoted x^*, is the **_optimal load_**, and the corresponding rate is the **_optimal power rate_** $\varphi^* = f(x^*)/x^*$. Using calculus we get the following observation.

Observation 1. *The optimal load is $x^* = (b/(\alpha - 1))^{1/\alpha}$. Additionaly, for any $x \neq x^*$, $f(x)/x > \varphi^*$.*

The following lemmas will be used in the analysis.

Lemma 1. *Consider two solutions $\pi = \{A_1, \ldots, A_m\}$ and $\pi' = \{A'_1, \ldots, A'_m\}$ of an instance of the VMA problem, such that for some $x, y \in [1, m]$ it holds that*

- $A_x \neq \emptyset$ *and* $A_y \neq \emptyset$;

- $A'_x = A_x \cup A_y$, $A'_y = \emptyset$, *and* $A_i = A'_i$, *for all* $i \neq x$ *and* $i \neq y$; *and*

- $\ell(A_x) + \ell(A_y) \leq \min\{x^*, C\}$.

Then, $P(\pi') < P(\pi)$.

Proof. Let $\ell(A_i) = x$ and $\ell(A_j) = y$. First we notice that π' is feasible because $x + y \leq C$. Now, using that $x + y \leq x^*$, we have

$$
\begin{aligned}
b &= (x^*)^\alpha(\alpha - 1) \geq (x + y)^\alpha(\alpha - 1) \\
&> (x + y)^\alpha \geq (x + y)^\alpha - (x^\alpha + y^\alpha)
\end{aligned}
\tag{2}
$$

where the second inequality comes from the fact that $\alpha > 1$. The above inequality is equivalent to

$$
2b + x^\alpha + y^\alpha > b + (x + y)^\alpha,
\tag{3}
$$

which implies the lemma. $\qquad\square$

---

to 1.5 and 3 and x^* values of $0.76C$ and $0.9C$ respectively.

From this lemma, it follows that the global power consumption can be reduced by having 2 VMs together in the same PM, when its aggregated load is smaller than $\min\{x^*, C\}$, instead of moving one VM to an unused PM. When we keep VMs together in a given partition we say that we are *using* Lemma 1.

Lemma 2. *Consider two solutions* $\pi = \{A_1, \ldots, A_m\}$ *and* $\pi' = \{A'_1, \ldots, A'_m\}$ *of an instance of the VMA problem, such that for some* $x, y \in [1, m]$ *it holds that*

- $A_x \cup A_y = A'_x \cup A'_y$, *while* $A_i = A'_i$, *for all* $x \neq i \neq y$;

- *none of* A_x, A_y, A'_x, *and* A'_y *is empty; and*

- $|\ell(A_x) - \ell(A_y)| < |\ell(A'_x) - \ell(A'_y)|$.

Then, $P(\pi) < P(\pi')$.

Proof. From the definition of $P(\cdot)$, to prove the claim is it enough to prove that $\ell(A_x)^\alpha + \ell(A_y)^\alpha < \ell(A'_x)^\alpha + \ell(A'_y)^\alpha$. Let us assume wlog that $\ell(A_x) \leq \ell(A_y)$ and $\ell(A'_x) \leq \ell(A'_y)$. Let us denote $L = \ell(A_x) + \ell(A_y) = \ell(A'_x) + \ell(A'_y)$, and assume that $\ell(A_x) = \delta_1 L$ and $\ell(A'_x) = \delta_2 L$. Note that $\delta_2 < \delta_1 \leq 1/2$. Then, the claim to be proven becomes

$$(\delta_1 L)^\alpha + ((1 - \delta_1)L)^\alpha < (\delta_2 L)^\alpha + ((1 - \delta_2)L)^\alpha \tag{4}$$

$$\delta_1^\alpha + (1 - \delta_1)^\alpha < \delta_2^\alpha + (1 - \delta_2)^\alpha \tag{5}$$

Which holds because the function $f(x) = x^\alpha + (1 - x)^\alpha$ is decreasing in the interval $(0, 1/2)$. $\square$

This lemma carries the intuition that balancing the load among the used PMs as much as possible reduces the power consumption.

Corollary 1. *Consider a solution* $\pi = \{A_1, \ldots, A_m\}$ *of an instance of the VMA problem with total load* $\ell(D)$, *such that exactly* k *of the* A_x *sets,* $x \in [1, m]$, *are non-empty (hence it uses* k *PMs). Then, the power consumption is lower bounded by the power of the (maybe unfeasible) solution that balances the load evenly, i.e.,*

$$P(\pi) \geq kb + k(\ell(D)/k)^\alpha. \tag{6}$$

3. Offline Analysis

3.1. NP-hardness

As was mentioned, it can be shown that deciding whether there is a feasible solution for an instance of the (C, m)-VMA problem is NP-complete or not, by a direct reduction from the 3-Partition problem. However, this result does not apply directly to the $(C, \cdot)$-VMA, $(\cdot, m)$-VMA, and $(\cdot, \cdot)$-VMA problems. We show now that these problems are NP-hard. We first prove the following lemma.

Lemma 3. *Given an instance of the VMA problem, any solution $\pi = \{A_1, \ldots, A_m\}$ where $\ell(A_i) \neq x^*$ for some $i \in [1, m] : A_i \neq \emptyset$, has power consumption $P(\pi) > \rho^* \ell(D) = \rho^* \sum_{d \in D} \ell(d)$.*

Proof. The total cost of π is $P(\pi) = \sum_{i \in [1,m]} f(\ell(A_i))$ which, from Observation 1, satisfies

$$
\begin{aligned}
P(\pi) \ &> \sum_{i \in [1,m]: A_i \neq \emptyset} \ell(A_i) \rho^* \\
&= \rho^* \sum_{i \in [1,m]: A_i \neq \emptyset} \sum_{d \in A_i} \ell(d) = \rho^* \sum_{d \in D} \ell(d).
\end{aligned}
\tag{7}
$$

$\square$

We show now in the following theorem that the different versions of the (C, m)-VMA problem with unbounded C or m are NP-hard.

Theorem 1. *The $(C, \cdot)$-VMA, $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems are strongly NP-hard, even if α is constant.*

Proof. We show a reduction from 3-Partition defined as follows [Garey and Johnson, 1979], which is strongly NP-complete.

INSTANCE: Set A of $3k$ elements, a bound $B \in \mathbb{Z}^+$ and, for each $a \in A$, a size $s(a) \in \mathbb{Z}^+$ such that $B/4 < s(a) < B/2$ and $\sum_{a \in A} s(a) = kB$.

QUESTION: can A be partitioned into k disjoint sets $\{A_1, A_2, \ldots, A_k\}$ such that $\sum_{a \in A_i} s(a) = B$ for each $1 \leq i \leq k$?

The reduction is as follows. Given an instance of 3-Partition on a set $A = \{a_1, \ldots, a_{3k}\}$ with bound B, and given a fixed value $\alpha > 1$, we define an instance $\mathcal{I}$ of $(\cdot, \cdot)$-VMA as follows: $D = \{a_1, \ldots, a_{3k}\}$, $\ell(\cdot) = s(\cdot)$, and $b = B^\alpha(\alpha-1)$ (i.e., $x^* = B$). (For the proof of the $(C, \cdot)$-VMA

12

and $(\cdot, m)$-VMA problems it is enough to set $C = B$ and $m = k$ when required.) We show now that the answer to the 3-Partition problem is YES if and only if the output $\pi = \{A_1, A_2, \ldots, A_m\}$ of the $(\cdot, \cdot)$-VMA problem on input $\mathcal{I}$ is such that $\sum_{i=1}^{m} f(\ell(A_i)) = kf(B)$.

For the direct implication, assume that there exists a partition $\{A_1, A_2, \ldots, A_k\}$ of A such that for each $i \in [1, k]$, $\sum_{a \in A_i} s(a) = B$. Then, in the context of the $(\cdot, \cdot)$-VMA problem, such partition has cost $\sum_{i=1}^{m} f(\ell(A_i)) = kf(B)$. We claim that any partition has at least cost $kf(B)$. In order to prove it, assume for the sake of contradiction that there is a partition $\pi' = \{A'_1, A'_2, \ldots, A'_m\}$ of $(\cdot, \cdot)$-VMA on input $\mathcal{I}$ with cost less than $kf(B)$. Then, there is some $i \in [1, m]$ such that $A'_i \neq \emptyset$ and $\ell(A'_i) \neq B$. From Lemma 3, $P(\pi') > \rho^* \ell(D) = (f(x^*)/x^*)kB$. Since $B = x^*$, we have that $P(\pi') > kf(B)$, which is a contradiction.

To prove the reverse implication, assume an output $\pi = \{A_1, A_2, \ldots, A_m\}$ of the $(\cdot, \cdot)$-VMA problem on input $\mathcal{I}$ such that $P(\pi) = \sum_{i=1}^{m} f(\ell(A_i)) = kf(B)$. Then, it must be $\forall i \in [1, m] : A_i \neq \emptyset, \ell(A_i) = B$. Otherwise, from Lemma 3, $P(\pi) > kf(B)$, a contradiction. $\qquad\square$

It is known that strongly NP-hard problems cannot have a fully polynomial-time approximation scheme (FPTAS) [Vazirani, 2004]. Hence, the following corollary.

Corollary 2. *The $(C, \cdot)$-VMA, $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems do not have fully polynomial-time approximation schemes (FPTAS), even if α is constant.*

In the following sections we show that, while the $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems have polynomial-time approximation schemes (PTAS), the $(C, \cdot)$-VMA problem cannot be approximated below $\frac{3}{2} \cdot \frac{\alpha - 1 + (2/3)^\alpha}{\alpha}$.

3.2. The $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA Problems Have PTAS

We have proved that the $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems are NP-hard in the strong sense and that, hence, there exists no FPTAS for them. However, Alon et al. [Alon et al., 1998], proved that if a function $f(\cdot)$ satisfies a condition denoted $F*$, then the problem of scheduling jobs in m identical machines so that $\sum_i f(M_i)$ is minimized has a PTAS, where M_i is the load of the jobs allocated to machine i. This result implies that if our function $f(\cdot)$ satisfies condition $F*$, the same PTAS can be used for the $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems. From Observation 6.1 in [Epstein and Sgall, 2004], it can be derived that, in fact, our power consumption function $f(\cdot)$ satisfies condition $F*$. Hence, the following theorem.

Theorem 2. *There are polynomial-time approximation schemes (PTAS) for the $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems.*

3.3. Bounds on the Approximability of the $(C, \cdot)$-VMA Problem

We study now the $(C, \cdot)$-VMA problem, where we consider an unbounded number of machines with bounded capacity C. We will provide a lower bound on its approximation ratio, independently on the relation between x^* and C; and upper bounds for the cases when $x^* \geq C$ and $x^* < C$.

3.3.1. Lower bound on the approximation ratio

The following theorem shows a lower bound on the approximation ratio of any offline algorithm for $(C, \cdot)$-VMA.

Theorem 3. *No algorithm achieves an approximation ratio smaller than $\frac{3}{2} \cdot \frac{\alpha - 1 + (\frac{2}{3})^\alpha}{\alpha}$ for the $(C, \cdot)$-VMA problem unless $\mathrm{P} = \mathrm{NP}$.*

Proof. The claim is proved showing a reduction from the partition problem [Garey and Johnson, 1979]. In the partition problem there is a set $A = \{a_1, a_2, \ldots, a_n\}$ of n elements, there is a size $s(a)$ for each element $a \in A$, and the sum $M = \sum_{a \in A} s(a)$ of the sizes of the elements in A. The problem decides whether there is a subset $A' \subset A$ such that $\sum_{a \in A'} s(a) = M/2$.

From an instance of the partition problem, we construct an instance of the $(C, \cdot)$-VMA problem as follows. The set of VMs in the system is $D = \{a_1, a_2, \ldots, a_n\}$, the load function is $\ell(\cdot) = s(\cdot)$, the capacity of each PM is set to $C = M/2$, and b is set to $b = C^\alpha(\alpha - 1)$ (i.e., $x^* = C$). Let us study the optimal partition π^* such that the total power consumption $P(\pi^*)$ is minimized. If there is a partition of D such that each subset in this partition has load $M/2$ then, from Observation 1, π^* has all the VMs assigned to two PMs. Otherwise, π^* needs at least 3 PMs to allocate all the VMs. From Corollary 1, the power consumption of this solution is lower bounded by the power of a (maybe unfeasible) partition that balances the load among the 3 PMs as evenly as possible. Formally,

$$\exists A' : \textstyle\sum_{a \in A'} s(a) = M/2$$
$$\Rightarrow \quad P(\pi^*) = 2b + 2\left(\tfrac{M}{2}\right)^\alpha = 2b + 2C^\alpha \tag{8}$$
$$\nexists A' : \textstyle\sum_{a \in A'} s(a) = M/2$$
$$\Rightarrow \quad P(\pi^*) \geq 3b + 3\left(\tfrac{M}{3}\right)^\alpha = 3b + 3\left(\tfrac{2C}{3}\right)^\alpha. \tag{9}$$

Comparing both values we obtain the following ratio.

$$\rho = \frac{3b + 3\left(\frac{2C}{3}\right)^\alpha}{2b + 2C^\alpha} = \frac{3C^\alpha(\alpha - 1) + 3\left(\frac{2C}{3}\right)^\alpha}{2C^\alpha(\alpha - 1) + 2C^\alpha}$$

$$= \frac{3}{2} \cdot \frac{\alpha - 1 + \left(\frac{2}{3}\right)^\alpha}{\alpha}. \tag{10}$$

Therefore, given any $\epsilon > 0$, having a polynomial-time algorithm $\mathcal{A}$ with approximation ratio $\rho - \epsilon$ would imply that this algorithm could be used to decide if there is a subset $A' \subset A$ such that $\sum_{a \in A'} s(a) = M/2$. In other words, this algorithm would be able to solve the partition problem. This contradicts the fact that the partition problem is NP-hard and no polynomial time algorithm solves it unless P $=$ NP. Therefore, there is no algorithm that achieves a $\rho - \epsilon = \frac{3}{2} \cdot \frac{\alpha - 1 + \left(\frac{2}{3}\right)^\alpha}{\alpha} - \epsilon$ approximation ratio for the $(C, \cdot)$-VMA problem unless P $=$ NP. $\qquad\square$

3.3.2. Upper bound on the approximation ratio for $x^* \geq C$

We study now an upper bound on the competitive ratio of the $(C, \cdot)$-VMA problem for the case when $x^* \geq C$. Under this condition, the best is to load each PM to its full capacity. Intuitively, an optimal solution should load every machine up to its maximum capacity or, if not possible, should balance the load among PMs to maximize the average load. The following lemma formalizes this observation.

Lemma 4. *For any system with unbounded number of PMs where $x^* \geq C$ the power consumption of the optimal assignment π^* is lower bounded by the power consumption of a (possibly not feasible) solution where $\ell(D)$ is evenly distributed among $\overline{m}$ PMs, where $\overline{m}$ is the minimum number of PMs required to allocate all VMs (i.e., the optimal solution of the packing problem). That is, $P(\pi^*) \geq \overline{m} \cdot b + \overline{m}(\ell(D)/\overline{m})^\alpha$.*

Proof. Denote the number of PMs used in an optimal $(C, \cdot)$-VMA solution π^* by m^*. By Corollary 1, we know that $P(\pi^*) \geq m^*b + m^*(\ell(D)/m^*)^\alpha$. Given that $\overline{m} \leq m^*$, we know that $\ell(D)/m^* \leq \ell(D)/\overline{m} \leq C \leq x^*$. Thus, for evenly-balanced loads the power consumption is reduced if the number of PMs is reduced, that is $m^*b + m^*(\ell(D)/m^*)^\alpha \geq \overline{m} \cdot b + \overline{m}(\ell(D)/\overline{m})^\alpha$. Hence, the claim follows. $\qquad\square$

Now we prove an upper bound on the approximation ratio showing a reduction to bin packing [Garey and Johnson, 1979]. The reduction works as follows. Let each PM be seen as a bin of

capacity C, and each VM be seen as an object to be placed in the bins, whose size is the VM load. Then, a solution for this bin packing problem instance yields a feasible (perhaps suboptimal) solution for the instance of $(C, \cdot)$-VMA. Moreover, using any bin-packing approximation algorithm, we obtain a feasible solution for $(C, \cdot)$-VMA that approximates the minimal number of PMs used. The power consumption of this solution approximates the power consumption of the optimal solution π^* of the instance of $(C, \cdot)$-VMA. In order to compute an upper bound on the approximation ratio of this algorithm, we will compare the power consumption of such solution against a lower bound on the power consumption of π^*. The following theorem shows the approximation ratio obtained.

Theorem 4. *For every $\epsilon > 0$, there exists an approximation algorithm for the $(C, \cdot)$-VMA problem when $x^* \geq C$ that achieves an approximation ratio of*

$$\rho < 1 + \epsilon + \frac{C^\alpha}{b} + \frac{1}{\overline{m}}, \tag{11}$$

where $\overline{m}$ is the minimum number of PMs required to allocate all the VMs.

Proof. Consider an instance of the $(C, \cdot)$-VMA problem. If $\ell(D) \leq C$, the optimal solution is to place all the VMs in one single PM. Hence, we assume in the rest of the proof that $\ell(D) > C$. Define the corresponding instance of bin packing following the reduction described above. Let the optimal number of bins to accommodate all VMs be $\overline{m}$. As shown in [Fernandez de la Vega and Lueker, 1981], for every $\epsilon > 0$, there is a polynomial-time algorithm that fits all VMs in $\widehat{m}$ bins, where $\widehat{m} \leq (1 + \epsilon)\overline{m} + 1$. From Lemma 2, once the number of PMs used $\widehat{m}$ is fixed, the power consumption is maximized when the load is unbalanced to the maximum. I.e., the power consumption of the assignment is at most $\widehat{m}b + (\ell(D)/C)C^\alpha$. On the other hand, as shown in Lemma 4, the power consumption of the optimal $(C, \cdot)$-VMA solution is at least $\overline{m} \cdot b + \overline{m} \left(\frac{\ell(D)}{\overline{m}} \right)^\alpha$. Then, we compute a bound on the approximation ratio as follows.

$$\rho \leq \frac{\widehat{m}b + \left(\frac{\ell(D)}{C} \right) C^\alpha}{\overline{m} \cdot b + \overline{m} \left(\frac{\ell(D)}{\overline{m}} \right)^\alpha} < \frac{\widehat{m}b + \left(\frac{\ell(D)}{C} \right) C^\alpha}{\overline{m} \cdot b + \overline{m} \left(\frac{C}{2} \right)^\alpha}, \tag{12}$$

where the second inequality comes from $\ell(D)/\overline{m} > C/2$. (If $\ell(D)/\overline{m} \leq C/2$, there must be two PMs whose loads add up to less than C, which contradicts the fact that $\overline{m}$ is the number of bins

16

used in the optimal solution of bin packing.) Let $\gamma = (x^*/C)^\alpha$. Then, replacing $b = \gamma C^\alpha(\alpha - 1)$, in Eq. (12) we have

$$
\begin{aligned}
\rho \;&<\; \frac{\widehat{m}\gamma C^\alpha(\alpha - 1) + \left(\frac{\ell(D)}{C}\right)C^\alpha}{\overline{m}\gamma C^\alpha(\alpha - 1) + \overline{m}\left(\frac{C}{2}\right)^\alpha} \\[2mm]
&=\; \frac{\widehat{m}\gamma(\alpha - 1) + \left(\frac{\ell(D)}{C}\right)}{\overline{m}\gamma(\alpha - 1) + \overline{m}\left(\frac{1}{2}\right)^\alpha} \;\leq\; \frac{\widehat{m}\gamma(\alpha - 1) + \overline{m}}{\overline{m}\gamma(\alpha - 1) + \left(\frac{\overline{m}}{2^\alpha}\right)} &\text{(13)} \\[2mm]
&\leq\; \frac{(\overline{m}(1 + \epsilon) + 1))\gamma(\alpha - 1) + \overline{m}}{\overline{m}\gamma(\alpha - 1) + \left(\frac{\overline{m}}{2^\alpha}\right)} &\text{(14)} \\[2mm]
&=\; \frac{(1 + \epsilon)\gamma(\alpha - 1) + 1}{\gamma(\alpha - 1) + \left(\frac{1}{2^\alpha}\right)} + \frac{\gamma(\alpha - 1)}{\overline{m}\gamma(\alpha - 1) + \left(\frac{\overline{m}}{2^\alpha}\right)} \\[2mm]
&=\; \frac{2^\alpha((1 + \epsilon)\gamma(\alpha - 1) + 1)}{2^\alpha\gamma(\alpha - 1) + 1} + \frac{2^\alpha\gamma(\alpha - 1)}{\overline{m}(2^\alpha\gamma(\alpha - 1) + 1)} \\[2mm]
&<\; \frac{(1 + \epsilon)\gamma(\alpha - 1) + 1}{\gamma(\alpha - 1)} + \frac{1}{\overline{m}} \\[2mm]
&=\; 1 + \epsilon + \frac{1}{\gamma(\alpha - 1)} + \frac{1}{\overline{m}} = 1 + \epsilon + \frac{C^\alpha}{b} + \frac{1}{\overline{m}} &\text{(15)}
\end{aligned}
$$

Inequality (13) follows from $\ell(D)/C \leq \overline{m}$, Inequality (14) from the approximation algorithm for bin packing, and the last inequality is because $\overline{m} > 0$. □

3.3.3. Upper bound on the approximation ratio for $x^* < C$

We study now the $(C, \cdot)$-VMA problem when $x^* < C$. In this case, the optimal load per PM is less than its capacity, so an optimal solution would load every PM to x^* if possible, or try to balance the load close to x^*. In this case we slightly modify the bin packing algorithm described above, reducing the bin size from C to x^*. Then, using an approximation algorithm for this bin packing problem, the following theorem can be shown.

Theorem 5. *For every $\epsilon > 0$, there exists an approximation algorithm for the $(C, \cdot)$-VMA problem when $x^* < C$ that achieves an approximation ratio of*

$$
\rho < \frac{\overline{m}}{m^*}\left((1 + \epsilon) + \frac{1}{\alpha - 1}\right) + \frac{1}{m^*}, \tag{16}
$$

where m^* is the number of PMs used by the optimal solution of $(C, \cdot)$-VMA, and $\overline{m}$ is the minimum number of PMs required to allocate all the VMs without exceeding load x^* (i.e., the optimal solution of the bin packing problem).

17

Proof. Consider an instance of the $(C, \cdot)$-VMA problem. If $\ell(D) \leq x^*$ then the optimal solution is to assign all the VMs to one single PM. Then, in the rest of the proof we assume that $\ell(D) > x^*$. Assuming m^* to be the number of PMs of an optimal $(C, \cdot)$-VMA solution π^* for load $\ell(D)$, from Corollary 1, we can claim that the power consumption $P(\pi^*)$ can be bounded as $P(\pi^*) \geq m^* b + m^* (\ell(D)/m^*)^\alpha$.

Now, let $\overline{m}$ be the minimum number of PMs required to allocate all the VMs of the $(C, \cdot)$-VMA problem without exceeding load x^*. As shown in [Fernandez de la Vega and Lueker, 1981], for every $\epsilon > 0$, there is a polynomial-time algorithm that fits all VMs in $\widehat{m}$ bins, where $\widehat{m} \leq (1 + \epsilon)\overline{m} + 1$. From Lemma 2, this approximation results in a power consumption no larger than $\widehat{m} b + (\ell(D)/x^*)(x^*)^\alpha$. Hence, the approximation ratio ρ of the solution obtained wit this algorithm can be bounded as follows.

$$\rho \leq \frac{\widehat{m} b + \left(\frac{\ell(D)}{x^*}\right)(x^*)^\alpha}{m^* b + m^* \left(\frac{\ell(D)}{m^*}\right)^\alpha}. \tag{17}$$

Since $\ell(D) > x^*$, we know that $\ell(D)/m^* > x^*/2$, since otherwise there are two used PMs whose load is no larger than x^*, contradicting by Lemma 1 the definition of m^*. Also, from the definition of $\overline{m}$, it follows that $\ell(D) \leq \overline{m} \cdot x^*$. Finally, recall that $b = (x^*)^\alpha(\alpha - 1)$. Applying these results to Eq. (17) we have the following.

$$
\begin{aligned}
\rho \;&<\; \frac{\widehat{m}(x^*)^\alpha(\alpha - 1) + \left(\frac{x^*\overline{m}}{x^*}\right)(x^*)^\alpha}{m^*(x^*)^\alpha(\alpha - 1) + m^*\left(\frac{x^*}{2}\right)^\alpha} \\[2mm]
&=\; \frac{\widehat{m}(\alpha - 1) + \overline{m}}{m^*(\alpha - 1) + m^*\left(\frac{1}{2}\right)^\alpha} \;\leq\; \frac{(\overline{m}(1 + \epsilon) + 1)(\alpha - 1) + \overline{m}}{m^*(\alpha - 1) + \frac{m^*}{2^\alpha}} \\[2mm]
&=\; \frac{\overline{m}(1 + \epsilon)(\alpha - 1) + \overline{m}}{m^*(\alpha - 1) + \frac{m^*}{2^\alpha}} + \frac{\alpha - 1}{m^*(\alpha - 1) + \frac{m}{2^\alpha}} \\[2mm]
&=\; \frac{\overline{m}}{m^*}\frac{2^\alpha((1 + \epsilon)(\alpha - 1) + 1)}{2^\alpha(\alpha - 1) + 1} + \frac{2^\alpha(\alpha - 1)}{2^\alpha m^*(\alpha - 1) + m^*} \\[2mm]
&\leq\; \frac{\overline{m}}{m^*}\left((1 + \epsilon) + \frac{1}{\alpha - 1}\right) + \frac{1}{m^*}, \tag{18}
\end{aligned}
$$

where the first inequality comes from applying the results aforementioned, and second one from using $\widehat{m} = \overline{m}(1 + \epsilon) + 1$, while the last one results from simplifying the previous equation. $\qquad\square$

4. Online Analysis

In this section, we study the online version of the VMA problem, i.e., when the VMs are revealed one by one. We first study lower bounds and then provide online algorithms and prove upper bounds on their competitive ratio.

4.1. Lower Bounds

In this section, we compute lower bounds on the competitive ratio for $(\cdot, \cdot)$-VMA, $(C, \cdot)$-VMA, $(\cdot, m)$-VMA, (C, m)-VMA and $(\cdot, 2)$-VMA problems. We start with one general construction that is used to obtain lower bounds on the first four cases. Then, we develop special constructions for $(\cdot, m)$-VMA and $(\cdot, 2)$-VMA that improve the lower bounds for these two problems.

4.1.1. General Construction

We prove lower bounds on the competitive ratio of $(\cdot, \cdot)$-VMA, $(C, \cdot)$-VMA, $(\cdot, m)$-VMA and (C, m)-VMA problems. These lower bounds are shown in the following two theorems. In Theorem 6, we prove a lower bound on the competitive ratio that is valid in the cases when C is unbounded and when it is larger or equal than x^*. The case $C \leq x^*$ is covered in Theorem 7.

Theorem 6. *There exists an instance of problems $(\cdot, \cdot)$-VMA, $(\cdot, m)$-VMA, $(C, \cdot)$-VMA and (C, m)-VMA when $C > x^*$, such that no online algorithm can guarantee a competitive ratio smaller than $\frac{(3/2)2^\alpha - 1}{2^\alpha - 1}$.*

Proof. We consider a scenario where, for any online algorithm, an adversary injects VMs of size ϵx^* ($\epsilon > 0$ is an arbitrarily small constant) to the system until the algorithm starts up a new PM. Let us assume that the total number of VMs injected is k. According to the adversary's behavior, the assignment of the VMs should be that all the VMs except one are allocated to a single PM while the second PM has only one VM. Depending on what the optimal solution is, we discuss the following two cases:

Case 1: $k \leq \frac{1}{\epsilon} \left(\frac{\alpha - 1}{1 - 2^{1-\alpha}} \right)^{1/\alpha}$. The optimal solution will allocate all the VMs to a single PM. Consequently, the competitive ratio of the online algorithm satisfies

$$\rho(k) \geq \lim_{\epsilon \to 0} \left(\frac{((k-1)\epsilon x^*)^\alpha + (\epsilon x^*)^\alpha + 2b}{(k\epsilon x^*)^\alpha + b} \right). \tag{19}$$

It can be easily verified that function $\rho(k)$ is monotone decreasing with k. That is, $\rho(k)$ is minimized when $k = \frac{1}{\epsilon}\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha}$. As a result, we obtain,

$$
\begin{aligned}
\rho(k) &\geq \lim_{\epsilon \to 0} \left(\frac{\left(\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + (\epsilon x^*)^\alpha + 2b}{\left(\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + b} \right) \\
&= \frac{\left(\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + 2(x^*)^\alpha(\alpha-1)}{\left(\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + (x^*)^\alpha(\alpha-1)} \\
&= \frac{3 - 2^{1-\alpha}}{2 - 2^{1-\alpha}} = \frac{(3/2)2^\alpha - 1}{2^\alpha - 1}.
\end{aligned}
\tag{20}
$$

Case 2: $k > \frac{1}{\epsilon}\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha}$. The optimal solution will use two PMs with $k/2$ PMs assigned to each PM. Accordingly, the competitive ratio of the online algorithm satisfies

$$
\rho(k) \geq \lim_{\epsilon \to 0} \left(\frac{((k-1)\epsilon x^*)^\alpha + (\epsilon x^*)^\alpha + 2b}{2\left(\frac{k\epsilon x^*}{2}\right)^\alpha + 2b} \right).
\tag{21}
$$

Similarly, we observe that $\rho(k)$ is monotone increasing with k. Consequently, the following inequality applies.

$$
\begin{aligned}
\rho(k) &\geq \lim_{\epsilon \to 0} \left(\frac{\left(\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + (\epsilon x^*)^\alpha + 2b}{2\left(\frac{1}{2}\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + 2b} \right) \\
&= \frac{\left(\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + 2(x^*)^\alpha(\alpha-1)}{2\left(\frac{1}{2}\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*\right)^\alpha + 2(x^*)^\alpha(\alpha-1)} \\
&= \frac{3 - 2^{1-\alpha}}{2 - 2^{1-\alpha}} = \frac{(3/2)2^\alpha - 1}{2^\alpha - 1}
\end{aligned}
\tag{22}
$$

Note that it can also happen that $C < \left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha} x^*$. In this case, k is smaller than $\frac{1}{\epsilon}\left(\frac{\alpha-1}{1-2^{1-\alpha}}\right)^{1/\alpha}$. Therefore, the competitive ratio is always larger than $\frac{(3/2)2^\alpha - 1}{2^\alpha - 1}$, proving the lower bound. $\qquad\square$

Theorem 7. *There exists an instance of problems $(C, \cdot)$-VMA and (C, m)-VMA when $C \leq x^*$ such that no online algorithm can guarantee a competitive ratio smaller than $(C^\alpha + 2b)/(b + \max(C^\alpha, 2(C/2)^\alpha + b))$.*

Proof. Similarly to the proof of Theorem 6, we prove the result by considering an adversarial injection of VMs of size ϵC. This injection stops when a new PM started up by an online algorithm.

We discuss the following two cases:

Case 1: $k \leq 1/\epsilon$. In this case, the optimal algorithm will assign all the VMs to a single PM. The competitive ratio of the online algorithm satisfies

$$\begin{aligned} \rho(k) &\geq \lim_{\epsilon \to 0} \frac{((k-1)\epsilon C)^\alpha + (\epsilon C)^\alpha + 2b}{(k\epsilon C)^\alpha + b} \\ &\geq \lim_{\epsilon \to 0} \frac{(1-\epsilon)^\alpha C^\alpha + 2b}{C^\alpha + b} \\ &\geq \frac{C^\alpha + 2b}{C^\alpha + b} \geq 2 - \frac{1}{\alpha} \end{aligned} \tag{23}$$

The second inequality results from applying $k \leq 1/\epsilon$, which is observed from the monotone decreasing property of function $\rho(k)$. The last inequality comes from computing the limit when ϵ goes to 0 and by applying $b \geq C^\alpha(\alpha - 1)$.

Case 2: $k > 1/\epsilon$. In this case, the adversary stops injecting VMs as there will be, mandatorily, two active PMs, one of them not capable to allocate more VMs and the second one hosting one single VM. Since all the VMs can not be consolidated to a single PM. The optimal solution would use also two PMs but evenly balancing the loads among them. The competitive ratio of the online algorithm satisfies

$$\begin{aligned} \rho(k) &= \lim_{\epsilon \to 0} \left(\frac{((k-1)\epsilon C)^\alpha + (\epsilon C)^\alpha + 2b}{2\left(\frac{k\epsilon C}{2}\right)^\alpha + 2b} \right) \\ &= \lim_{\epsilon \to 0} \left(\frac{C^\alpha + (\epsilon C)^\alpha + 2b}{2\left(\frac{C+\epsilon C}{2}\right)^\alpha + 2b} \right) = \frac{C^\alpha + 2b}{2\left(\frac{C}{2}\right)^\alpha + 2b}. \end{aligned} \tag{24}$$

Hence, combining the results from both cases 1 and 2 we obtain the bound presented in Theorem 7. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

4.1.2. Special Constructions for $(\cdot, m)$-VMA and $(\cdot, 2)$-VMA

We show first that for m PMs there is a lower bound on the competitive ratio that improves the previous lower bound when $\alpha > 4.5$. Secondly, we prove a particular lower bound for problem $(\cdot, 2)$-VMA, that improves the previous lower bound when $\alpha > 3$.

Theorem 8. *There exists an instance of problem $(\cdot, m)$-VMA such that no online algorithm can guarantee a competitive ratio smaller than $3^\alpha / (2^{\alpha+2} + \epsilon)$ for any $\epsilon > 0$.*

Proof. We prove the result by giving an adversarial arrival of VMs. We evaluate the competitive ratio of any online algorithm ALG with respect to an algorithm OPT that distributes the VMs among all the PMs "as evenly as possible". We define a value $\beta > 1$ such that $\epsilon \geq (\alpha - 1)/\beta^\alpha$ for some value $\epsilon > 0$. Note that such value β can be defined for any $\epsilon > 0$. The adversarial arrival follows. In a first phase, m VMs arrive, each with load βx^*.

Let π be the partition given by ALG. We show first that if π uses less than $3m/4$ PMs[5] or some PM is assigned more than 2 VMs there exists another partition that can be obtained from π, it uses exactly $3m/4$ PMs, no PM is assigned more than 2 VMs, and the power consumption is not worse.

If π uses less than $3m/4$ PMs, then there exists another partition π' that uses exactly $3m/4$ PMs with a power consumption that is not worse than $P(\pi)$. To see why, notice that there are PMs in π that are assigned more than one VM and that each load is $\beta x^* > x^*$. Then, applying repeatedly Lemma 1 until $3m/4$ PMs are used, where ℓ_1 and ℓ_2 are the loads of any pair of VMs assigned to the same PM, a partition π' such that $P(\pi') \leq P(\pi)$ can be obtained.

If in π' some PM is assigned more than 2 VMs, then there exists another partition π'' where no PM is assigned more than 2 VMs with a power consumption that is not worse than $P(\pi')$. To see why, consider the following reassignment procedure. Repeatedly until there is no such PM, locate a PM s_i with at least 3 VMs. Then, locate a PM s_j with one single VM (which exists by the pigeonhole principle). Then, move one VM from s_i to s_j. From Lemma 2 each movement decreases the power consumed. Hence, π'' is still a partition that uses $3m/4$ PMs, each PM has at most 2 VMs assigned, and $P(\pi'') \leq P(\pi')$.

Then, we know that $P(\pi)$ is not smaller than the power consumption of a partition where exactly $3m/4$ PMs are used and no PM is assigned more than 2 VMs. On the other hand, OPT would have assigned each VM to a different PM. Thus, using that $x^* = (b/(\alpha - 1))^{1/\alpha}$, the competitive ratio is

$$
\begin{aligned}
\rho \quad &\geq \quad \frac{(2\beta x^*)^\alpha m/4 + (\beta x^*)^\alpha m/2 + 3mb/4}{m(\beta x^*)^\alpha + mb} \\
&\geq \quad \frac{(2^{\alpha-2} + 1/2)\beta^\alpha}{\beta^\alpha + (\alpha - 1)} \geq 2^{\alpha-3} + 1/4,
\end{aligned}
\tag{25}
$$

---

[5]For clarity we omit floors and ceilings in the proof.

where the last inequality follows from $\beta^\alpha \geq (\alpha - 1)$. Finally, observe that $2^{\alpha-3} + 1/4 \geq 3^\alpha/(2^{\alpha+2} + \epsilon)$ for $\alpha > 1$. No more VMs arrive in this case.

Let us consider now the the case where ALG assigns the m initial VMs to more than $3m/4$ PMs. Then, after ALG has assigned the first m VMs, a second batch of $m/2$ VMs arrive, each VM with load $2\beta x^*$. Let π be the partition output by ALG after this second batch is assigned. If in π two of the second batch VMs are assigned to the same PM s_i, by the pigeonhole principle there is at least one PM s_j with at most load βx^*. Then, from Lemma 2, the power consumed is reduced if one of the new VMs is moved from s_i to s_j. After repeating this process as many times as possible, a partition π' is obtained where each of the VMs of the second batch is assigned to a different PM, and $P(\pi') \leq P(\pi)$. Since ALG used more than $3m/4$ PMs in the first batch, in π', there are at least $m/4$ PMs with load $3\beta x^*$. On the other hand, OPT can distribute all the VMs in such a way that each PM has a load of $2\beta x^*$. Thus, the bound on the competitive ratio is as follows.

$$\rho \geq \frac{m(3\beta x^*)^\alpha/4}{m(2\beta x^*)^\alpha + mb} \geq \frac{3^\alpha}{2^{\alpha+2} + \epsilon}, \tag{26}$$

where the last inequality follows from $\epsilon \geq (\alpha - 1)/\beta^\alpha$. $\qquad \square$

Now, we show a stronger lower bound on the competitive ratio for $(\cdot, 2)$-VMA problem.

Theorem 9. *There exists an instance of problem $(\cdot, 2)$-VMA such that no online algorithm can guarantee a competitive ratio smaller than $3^\alpha/2^{\alpha+1}$.*

Proof. We prove the result by showing an adversarial arrival of VM. We evaluate the competitive ratio of any online algorithm ALG with respect to an optimal algorithm OPT that knows the future VM arrivals. The adversarial arrival follows. In a first phase two VM d_1 and d_2 arrive, with loads $\ell(d_1) = \ell(d_2) = 6x^*$ (Recall from Section 2 that $x^* = (b/(\alpha - 1))^{1/\alpha}$).

If ALG assigns both VMs to the same PM, the power consumed will be $(12x^*)^\alpha + b$, whereas OPT would assign them to different PMs, with a power consumption of $2((6x^*)^\alpha + b)$. Hence, the ratio ρ would be

$$\begin{aligned} \rho &= \frac{(12x^*)^\alpha + b}{2((6x^*)^\alpha + b)} > \frac{12^\alpha}{2(6^\alpha + \alpha - 1)} \\ &> \frac{12^\alpha}{2(6^\alpha + 2^\alpha)} = \frac{6^\alpha}{2(3^\alpha + 1)}, \end{aligned} \tag{27}$$

where the first inequality follows from $\alpha > 1$ and the second from $\alpha - 1 < 2^\alpha$ for any $\alpha > 1$. It is enough to prove that $6^\alpha/(2(3^\alpha + 1)) \geq (3/2)^\alpha /2$, or equivalently $4^\alpha \geq 3^\alpha + 1$, which is true for any $\alpha > 1$. Then, there are no new VM arrivals.

If, otherwise, ALG assigns each VM d_1 and d_2 to a different PM, then a third VM d_3 arrives, with load $\ell(d_3) = 12x^*$. Then, ALG must assign it to one of the PMs. Independently of which PM is used, the power consumption of the final configuration is $(18x^*)^\alpha + (6x^*)^\alpha + 2b$. On its side, OPT assigns d_1 and d_2 to one PM, and d_3 to the other, with a power consumption of $2((12x^*)^\alpha + b)$. Hence, the competitive ratio ρ is

$$
\begin{aligned}
\rho \;=\; & \frac{(18x^*)^\alpha + (6x^*)^\alpha + 2b}{2((12x^*)^\alpha + b)} > \frac{18^\alpha + 6^\alpha}{2(12^\alpha + \alpha - 1)} \\
> \;& \frac{18^\alpha + 6^\alpha}{2(12^\alpha + 4^\alpha)} \geq (3/2)^\alpha /2,
\end{aligned}
\tag{28}
$$

where the first inequality follows from $\alpha > 1$, the second from $\alpha - 1 < 4^\alpha$ for any $\alpha > 1$, and the third from $(9^\alpha + 3^\alpha)/(6^\alpha + 2^\alpha) \geq (3/2)^\alpha$, what can be checked to be true. Then, there are no new VM arrivals and the claim follows. $\square$

4.2. Upper Bounds

Now, we study upper bounds for $(\cdot, \cdot)$-VMA, $(C, \cdot)$-VMA, and $(\cdot, 2)$-VMA problems. We start giving an online VMA algorithm that can be used in $(\cdot, \cdot)$-VMA and $(C, \cdot)$-VMA problems. The algorithm uses the load of the new revealed VM in order to decide the PM where it will be assigned. If the load of the revealed VM is strictly larger than $\min\{x^*, C\}/2$, the algorithm assigns this VM to a new PM without any other VM already assigned to it. Otherwise, the algorithm schedules the revealed VM to any loaded PM whose current load is smaller or equal than $\frac{\min\{x^*, C\}}{2}$. Hence, when this new VM is assigned, the load of this PM remains smaller than $\min\{x^*, C\}$. If there is no such loaded PM, the revealed VM is assigned to a new PM. Note that, since the case under consideration assumes the existence of an unbounded number of PMs, there exists always one new PM. A detailed description of this algorithm is shown in Algorithm 1. As before, A_j denotes the set of VMs assigned to PM s_j at a given time.

We prove the approximation ratio of Algorithm 1 in the following two theorems.

Theorem 10. *There exists an online algorithm for $(\cdot, \cdot)$-VMA and $(C, \cdot)$-VMA when $x^* < C$ that*

Algorithm 1: Online algorithm for $(\cdot, \cdot)$-VMA and $(C, \cdot)$-VMA problems.
---

for *each VM d_i* **do**

> **if** $\ell(d_i) > \frac{\min\{x^*, C\}}{2}$ **then**
> > | d_i is assigned to a new PM
>
> **else**
> > | d_i is assigned to any loaded PM s_j where $\ell(A_j) \leq \frac{\min\{x^*, C\}}{2}$. If such loaded PM does
> > | not exist, d_i is assigned to a new PM

achieves the following competitive ratio:

$$\rho \ = 1, \ \textit{if no VM } d_i \textit{ has load such that } \ell(d_i) < x^*, \tag{29}$$

$$\rho \ \leq \left(1 - \frac{1}{\alpha}\left(1 - \frac{1}{2^\alpha}\right)\right)\left(2 + \frac{x^*}{\ell(D_s)}\right), \ \textit{otherwise.} \tag{30}$$

Proof. We proceed with the analysis of the competitive ratio of Algorithm 1 shown above. Let us first consider an optimal algorithm, that is, an algorithm that gives an optimal solution for any instance. Let us denote by π^* the optimal solution obtained by the optimal algorithm, and A_i the load assigned to PM s_i in that solution, for a particular instance of VMA problem. Furthermore, load A_i is decomposed in $d_{i_1}, d_{i_2}, \ldots, d_{i_{k_i}}$, where each d_{i_j} is a VM that π^* assigns to s_i. Using simple algebra, it holds:

$$f(\ell(A_i)) = \frac{f(\ell(A_i))}{\ell(A_i)}(\ell(d_{i_1}) + \ell(d_{i_2}) + \cdots + \ell(d_{i_{k_i}})). \tag{31}$$

It is possible now to split the set A_i in two sets, one with those VMs assigned to s_i whose load is strictly smaller than x^* and a second set that contains those VMs assigned to s_i whose load is bigger than x^*. In terms of notation, we say that A_i is split in B_i and S_i (where B stands for Big loads and S stands for Small loads). Therefore, it also holds:

$$f(\ell(A_i)) = \sum_{d_{i_j} \in B_i} \frac{f(\ell(A_i))}{\ell(A_i)}\ell(d_{i_j}) + \sum_{d_{i_j} \in S_i} \frac{f(\ell(A_i))}{\ell(A_i)}\ell(d_{i_j}). \tag{32}$$

On the other hand, by definition of x^*, it holds that: $f(\ell(A_i))/\ell(A_i) \geq f(x^*)/x^*$ for all i (indeed, for any load). Moreover, if a PM has been assigned with a load $\ell(d_{i_j})$ bigger than x^*, it also holds that $f(\ell(A_i))/\ell(A_i) \geq f(\ell(d_{i_j}))/\ell(d_{i_j})$. Hence, we obtain the following inequality:

$$f(\ell(A_i)) \geq \sum_{d_{i_j} \in B_i} f(\ell(d_{i_j})) + \sum_{d_{i_j} \in S_i} \frac{f(x^*)}{x^*}\ell(d_{i_j}). \tag{33}$$

25

In order to lower bound the power consumption of the solution π^*, we plug the above inequality into the corresponding equation:

$$
\begin{aligned}
P(\pi^*) &= \sum_{A_i \neq \emptyset} f(\ell(A_i)) \\
&\geq \sum_{A_i \neq \emptyset} \sum_{d_{i_j} \in B_i} f(\ell(d_{i_j})) + \frac{f(x^*)}{x^*} \sum_{A_i \neq \emptyset} \sum_{d_{i_j} \in S_i} \ell(d_{i_j}),
\end{aligned}
\tag{34}
$$

or, equivalently expressed in more compact notation:

$$
P(\pi^*) \geq \sum_{d_i : \ell(d_i) \geq x^*} f(\ell(d_i)) + \frac{f(x^*)}{x^*} \sum_{d_i : \ell(d_i) < x^*} \ell(d_i).
\tag{35}
$$

Consider now Algorithm 1. Let us denote by π a solution that Algorithm 1 gives for a particular instance. Also, let us denote by $\hat{A}_i$ the load assigned by Algorithm 1 to PM s_i. Note that due to the design of the algorithm, after the last VM has been assigned, either there is only one loaded PM whose current load is smaller than $x^*/2$, or every loaded PM has a load at least $x^*/2$. We study these two cases separately.

Case 1: $\ell(\hat{A}_i) \geq x^*/2$ for all i. In this case, in a solution provided by π there are PMs with two types of load: those that are loaded with one VM whose load is no smaller than x^*, and those that are loaded with VMs whose load is strictly smaller than x^*, nonetheless, their total load is bigger than $x^*/2$. Note that due to the design of the algorithm, none of the PMs in the second group has a load bigger than x^*. Let us denote by B the set of VMs with load at least x^*, and D_s the set of VMs with load less than x^*. Therefore, it holds:

$$
\begin{aligned}
P(\pi) &= \sum_{d \in B} f(\ell(d)) + \sum_{\frac{x^*}{2} \leq \ell(\hat{A}_i) \leq x^*} f(\ell(\hat{A}_i)) \\
&\leq \sum_{d \in B} f(\ell(d)) + \frac{f(\frac{x^*}{2})}{\frac{x^*}{2}} \ell(D_s).
\end{aligned}
\tag{36}
$$

Computing the ratio ρ between $P(\pi)$ and $P(\pi^*)$, we obtain the following inequality:

$$
\begin{aligned}
\rho &\leq \frac{\sum_{d \in B} f(\ell(d)) + \frac{f(\frac{x^*}{2})}{\frac{x^*}{2}} \ell(D_s)}{\sum_{d \in B} f(\ell(d)) + \frac{f(x^*)}{x^*} \ell(D_s)} \leq \frac{\frac{f(\frac{x^*}{2})}{\frac{x^*}{2}} \ell(D_s)}{\frac{f(x^*)}{x^*} \ell(D_s)} \\
&= 2 \frac{f(\frac{x^*}{2})}{f(x^*)} = 2 \left(1 - \frac{1}{\alpha} \left(1 - \frac{1}{2^\alpha} \right) \right).
\end{aligned}
\tag{37}
$$

Case 2: there exists s_i such that $\ell(\hat{A}_i) < x^*/2$. In this case, π gives solutions with three types of loaded PMs: those that are loaded with one VM whose load is bigger than x^*, those that are loaded with VMs whose load is strictly smaller than x^*, but which total load is at least $x^*/2$, and one PM whose total load is is strictly smaller than $x^*/2$. Let us denote such a PM by s'. Therefore, it holds:

$$
\begin{aligned}
P(\pi) &= \sum_{d \in B} f(\ell(d)) + \sum_{\frac{x^*}{2} \leq \ell(\hat{A}_i) \leq x^*} f(\ell(\hat{A}_i)) + f(\ell(\hat{A}_{s'})) \\
&\leq \sum_{d \in B} f(\ell(d)) + \frac{f(\frac{x^*}{2})}{\frac{x^*}{2}} \Big(\ell(D_s) - \ell(\hat{A}_{s'}) \Big) + f(\ell(\hat{A}_{s'})) \\
&= \sum_{d \in B} f(\ell(d)) + \frac{f(\frac{x^*}{2})}{\frac{x^*}{2}} \Big(\ell(D_s) - \ell(\hat{A}_{s'}) \Big) + \ell(\hat{A}_{s'})^\alpha + b.
\end{aligned}
\tag{38}
$$

Let us denote the latter expression by $\Pi(\pi)$. Computing the ratio ρ between $P(\pi)$ and $P(\pi^*)$, we obtain the following inequality:

$$
\begin{aligned}
\rho &\leq \frac{\Pi(\pi)}{\sum_{d \in B} f(\ell(d)) + \frac{f(x^*)}{x^*} \ell(D_s)} \\
&\leq 2 \left(1 - \frac{1}{\alpha} \left(1 - \frac{1}{2^\alpha} \right) \right) + \frac{\ell(\hat{A}_{s'})^\alpha - \ell(\hat{A}_{s'}) \frac{f(\frac{x^*}{2})}{\frac{x^*}{2}} + b}{\frac{f(x^*)}{x^*} \ell(D_s)} \\
&\leq 2 \left(1 - \frac{1}{\alpha} \left(1 - \frac{1}{2^\alpha} \right) \right) + \frac{\ell(\hat{A}_{s'})^\alpha + b}{\frac{f(x^*)}{x^*} \ell(D_s)} \\
&\leq 2 \left(1 - \frac{1}{\alpha} \left(1 - \frac{1}{2^\alpha} \right) \right) + \frac{(\frac{x^*}{2})^\alpha + b}{\frac{f(x^*)}{x^*} \ell(D_s)} \\
&= \left(1 - \frac{1}{\alpha} \left(1 - \frac{1}{2^\alpha} \right) \right) \left(2 + \frac{x^*}{\ell(D_s)} \right).
\end{aligned}
\tag{39}
$$

Since $x^*/\ell(D_s)$ is always positive, the competitive ratio of Algorithm 1 is equal to $2^{\alpha-1} + x^*/\ell(D_s)$. Observe that, when no VM d has load $\ell(d) < x^*$, i,e., $S = \emptyset$, $P(\pi)$ and $P(\pi^*)$ are equal. Hence, the competitive ratio is 1. $\qquad\square$

Theorem 11. *There exists an online algorithm for $(C, \cdot)$-VMA when $x^* \geq C$ that achieves competitive ratio $\rho \leq \frac{2b}{C} \left(1 + \frac{1}{(\alpha-1)2^\alpha} \right) \left(2 + \frac{C}{\ell(D)} \right)$.*

Proof. We proceed with the analysis of the competitive ratio of Algorithm 1 in the case when $x^* \geq C$. The analysis uses the same technique used in the proof for the previous theorem. Hence, we just show the difference.

On the one hand, when $x^* \geq C$, it holds that $f(\ell(A_i))/\ell(A_i) \geq f(C)/C$ due to the fact that $f(x)/x$ is monotone decreasing in interval $(0, C]$. It is also obvious that all the PMs will be loaded no more C. As a result, the optimal power consumption for $(C, \cdot)$-VMA can be bounded by

$$P(\pi^*) \geq \frac{f(C)}{C}\ell(D). \tag{40}$$

On the other hand, the solution given by Algorithm 1 can also be upper bounded. We consider the following two cases.

Case 1: $\ell(\hat{A}_i) \geq C/2$ for all i. In this case, every PM will be loaded between $C/2$ and C. Consequently,

$$P(\pi) = \sum_{\frac{C}{2} \leq \ell(\hat{A}_i) \leq C} f(\ell(\hat{A}_i)) \leq \frac{f(\frac{C}{2})}{\frac{C}{2}}\ell(D). \tag{41}$$

The competitive ratio ρ then satisfies

$$\rho \leq \frac{\frac{f(\frac{C}{2})}{\frac{C}{2}}\ell(D)}{\frac{f(C)}{C}\ell(D)} = 2\frac{f(\frac{C}{2})}{f(C)} \leq \frac{2b}{C}\left(1 + \frac{1}{(\alpha - 1)2^\alpha}\right). \tag{42}$$

Case 2: there exists s_i such that $\ell(\hat{A}_i) < C/2$. In this case, it holds:

$$\begin{aligned}
P(\pi) &= \sum_{\frac{C}{2} \leq \ell(\hat{A}_i) \leq C} f(\ell(\hat{A}_i)) + f(\ell(\hat{A}_{s'})) \\
&\leq \frac{f(\frac{C}{2})}{\frac{C}{2}}\left(\sum_{d_i:\ell(d_i) \leq C} \ell(d_i) - \ell(\hat{A}_{s'})\right) + f(\ell(\hat{A}_{s'})) \\
&= \frac{f(\frac{C}{2})}{\frac{C}{2}}\left(\ell(D) - \ell(\hat{A}_{s'})\right) + \ell(\hat{A}_{s'})^\alpha + b.
\end{aligned} \tag{43}$$

The competitive ratio ρ then satisfies

$$\begin{aligned}
\rho &\leq \frac{P(\pi)}{\frac{f(C)}{C}\ell(D)} \leq \frac{2b}{C}\left(1 + \frac{1}{(\alpha - 1)2^\alpha}\right) + \\
&\quad + \frac{\ell(\hat{A}_{s'})^\alpha - \ell(\hat{A}_{s'})\frac{f(\frac{C}{2})}{\frac{C}{2}} + b}{\frac{f(C)}{C}\ell(D)} \\
&\leq \frac{2b}{C}\left(1 + \frac{1}{(\alpha - 1)2^\alpha}\right) + \frac{\ell(\hat{A}_{s'})^\alpha + b}{\frac{f(C)}{C}\ell(D)} \\
&\leq \frac{2b}{C}\left(1 + \frac{1}{(\alpha - 1)2^\alpha}\right) + \frac{(\frac{C}{2})^\alpha + b}{\frac{f(C)}{C}\ell(D)} \\
&= \frac{2b}{C}\left(1 + \frac{1}{(\alpha - 1)2^\alpha}\right)\left(2 + \frac{C}{\ell(D)}\right).
\end{aligned} \tag{44}$$

28

□

4.2.1. Upper Bounds for $(\cdot, 2)$-VMA problem

We now present an algorithm (detailed in Algorithm 2) for $(\cdot, 2)$-VMA problem and show an upper bound on its competitive ratio. A_1 and A_2 are the sets of VMs assigned to PMs s_1 and s_2, respectively, at any given time.

Algorithm 2: Online algorithm for $(\cdot, 2)$-VMA.

---

for *each VM d_i* **do**

    if $\ell(d_i) + \ell(A_1) \leq (b/(2^\alpha - 2))^{1/\alpha}$ **or** $\ell(A_1) \leq \ell(A_2)$ **then**

        d_i is assigned to s_1;

    else

        d_i is assigned to s_2;

We prove the approximation ratio of Algorithm 2 in the following theorem.

Theorem 12. *There exists an online algorithm for $(\cdot, 2)$-VMA that achieves the following competitive ratios.*

$$\rho = 1, \quad \text{for } \ell(D) \leq \left(\tfrac{b}{2^\alpha - 2}\right)^{1/\alpha}, \tag{45}$$

$$\rho \leq \max\left\{2, \left(\frac{3}{2}\right)^{\alpha-1}\right\}, \quad \text{for } \ell(D) > \left(\tfrac{b}{2^\alpha - 2}\right)^{1/\alpha}. \tag{46}$$

Proof. Consider Algorithm 2 shown above. If $\ell(D) \leq (b/(2^\alpha - 2))^{1/\alpha}$, then the competitive ratio is 1 as we show. Algorithm 2 assigns all the VMs to PM s_1. On the other hand, the optimal offline algorithm also assigns all the VMs to one PM. To prove it, it is enough to show that $\ell(D)^\alpha + b < \ell(A_1)^\alpha + \ell(A_2)^\alpha + 2b$. Using that $\ell(A_1)^\alpha + \ell(A_2)^\alpha > 2(\ell(D)/2)^\alpha$ and manipulating, it is enough to prove $\ell(D) < 2(b/(2^\alpha - 2))^{1/\alpha}$. This is true for $\ell(D) \leq (b/(2^\alpha - 2))^{1/\alpha}$.

We consider now the case $(b/(2^\alpha - 2))^{1/\alpha} < \ell(D) < 2(b/(2^\alpha - 2))^{1/\alpha}$. Within this range, for the optimal algorithm is still better to assign all VMs to one PM, as shown. Then, the competitive ratio ρ is

$$\rho = \frac{\ell(A_1)^\alpha + \ell(A_2)^\alpha + 2b}{\ell(D)^\alpha + b} \leq \frac{\ell(D)^\alpha + 2b}{\ell(D)^\alpha + b} < 2. \tag{47}$$

29

Consider any given step after $\ell(D) \geq 2 \left(b/(2^{\alpha} - 2) \right)^{1/\alpha}$. Within this range, the optimal algorithm may assign the VMs to one or both PMs. If the optimal algorithm assigns to one PM, Inequality 47 applies. Otherwise, the competitive ratio ρ is

$$
\begin{aligned}
\rho &= \frac{\ell(A_1)^{\alpha} + \ell(A_2)^{\alpha} + 2b}{2(\ell(D)/2)^{\alpha} + 2b} \leq 2^{\alpha-1} \frac{\ell(A_1)^{\alpha} + \ell(A_2)^{\alpha}}{\ell(D)^{\alpha}} \\
&= 2^{\alpha-1} \frac{\ell(A_1)^{\alpha}/\ell(A_2)^{\alpha} + 1}{(\ell(A_1)/\ell(A_2) + 1)^{\alpha}}.
\end{aligned}
\tag{48}
$$

Then, in order to obtain a ratio at most $x^{\alpha}/2$, where x will be set later, it is enough to guarantee

$$
\begin{aligned}
2^{\alpha-1} \frac{\ell(A_1)^{\alpha}/\ell(A_2)^{\alpha} + 1}{(\ell(A_1)/\ell(A_2) + 1)^{\alpha}} &\leq \frac{x^{\alpha}}{2} \\
\frac{(\ell(A_1)/\ell(A_2))^{\alpha} + 1}{(\ell(A_1)/\ell(A_2) + 1)^{\alpha}} &\leq \left(\frac{x}{2}\right)^{\alpha}.
\end{aligned}
\tag{49}
$$

Without loss of generality, assume $\ell(A_1) \leq \ell(A_2)$. This implies that $(\ell(A_1)/\ell(A_2))^{\alpha} \leq \ell(A_1)/\ell(A_2)$. Then, it is enough to have

$$
\frac{\ell(A_1)/\ell(A_2) + 1}{(\ell(A_1)/\ell(A_2) + 1)^{\alpha}} \leq \left(\frac{x}{2}\right)^{\alpha}.
\tag{50}
$$

Let us now define $\ell(A_1) + \ell = \ell(A_2)$ for some $\ell \geq 0$. Manipulating and replacing, it is enough to show

$$
\frac{\ell}{\ell(A_1)} \leq \frac{2 - (2/x)^{\alpha/(\alpha-1)}}{(2/x)^{\alpha/(\alpha-1)} - 1}.
\tag{51}
$$

If Inequality 51 holds the theorem is proved. Otherwise, the following claim is needed.

Claim 1. *If $\ell(D) \geq 2 \left(b/(2^{\alpha} - 2) \right)^{1/\alpha}$, then there must exist a VM d_i in D such that $\ell(d_i) \geq |\ell(A_2) - \ell(A_1)|$.*

Proof. If $\ell(A_2) = \ell(A_1)$ the claim follows trivially. Assume that $\ell(A_2) \neq \ell(A_1)$. Consider any given time when $\ell(D) \geq 2 \left(b/(2^{\alpha} - 2) \right)^{1/\alpha}$. For the sake of contradiction, assume that for all $d_i \in D$ it is $\ell(d_i) < |\ell(A_2) - \ell(A_1)|$. Let $d_1, d_2, \ldots, d_r$ be the order in which the VMs were revealed to Algorithm 2. And let the respective sets of VMs be called $D_i = \{d_j | j \in [1, i]\}$, that is $D_r = D$. Given that $\ell(D) \geq 2 \left(b/(2^{\alpha} - 2) \right)^{1/\alpha} > \left(b/(2^{\alpha} - 2) \right)^{1/\alpha}$, the VM d_r was assigned to the PM with smaller load. Then, either $\ell(d_r) \geq |\ell(A_2) - \ell(A_1)|$ which would be a contradiction, or if $\ell(d_r) < |\ell(A_2) - \ell(A_1)|$ the PM with the smaller load before and after assigning d_r is the

same. The argument can be repeated iteratively backwards for each d_{r-1}, d_{r-2}, etc. until, for some $j \in [1, r)$, either it is $\ell(d_j) \geq |\ell(A_2) - \ell(A_1)|$ reaching a contradiction, or the total load is $\ell(D_j) < (b/(2^\alpha - 2))^{1/\alpha}$. If the latter is the case, we know that for $i \in [1, j]$ every d_i was assigned to s_1. Recall that for $i \in (j, r]$ each d_i was assigned to the same PM. And, given that d_{j+1} is the first VM for which the total load is at least $(b/(2^\alpha - 2))^{1/\alpha}$, that PM is s_2. But then, we have $\ell(A_2) < \ell(A_1) < (b/(2^\alpha - 2))^{1/\alpha}$, which is a contradiction with the assumption that $\ell(D) \geq 2 (b/(2^\alpha - 2))^{1/\alpha}$. $\qquad\qquad\square$

Using Claim 1 we know that there exists a d_i in the input such that

$$\ell(d_i) \geq \ell > \ell(A_1) \frac{2 - (2/x)^{\alpha/(\alpha-1)}}{(2/x)^{\alpha/(\alpha-1)} - 1}. \qquad (52)$$

From the latter, it can be seen that if $x \geq 2(3/4)^{\frac{\alpha-1}{\alpha}}$, then we have that $\ell > 2\ell(A_1)$. Then, the competitive ratio ρ is

$$\begin{aligned} \rho &= \frac{\ell(A_1)^\alpha + (\ell(A_1) + \ell)^\alpha + 2b}{(2\ell(A_1))^\alpha + \ell^\alpha + 2b} \\ &\leq \frac{\ell(A_1)^\alpha + (\ell(A_1) + \ell)^\alpha}{(2\ell(A_1))^\alpha + \ell^\alpha}. \end{aligned} \qquad (53)$$

Using calculus, this ratio is maximized for $\ell = 2\ell(A_1)$ for $\ell \geq 2\ell(A_1)$. Then, we have $\rho \leq (1 + 3^\alpha)/(2 \cdot 2^\alpha)$. Then, in order to obtain a ratio at most $x^\alpha/2$, it is enough to guarantee $(1 + 3^\alpha)/(2 \cdot 2^\alpha) \leq x^\alpha/2$ which yields $x \geq ((1 + 3^\alpha)/2^\alpha)^{1/\alpha}$.

Given that, for any $\alpha \geq 1$, it holds:

$$2(3/4)^{1-1/\alpha} \geq ((1 + 3^\alpha)/2^\alpha)^{1/\alpha}. \qquad (54)$$

Then, the competitive ratio is $\rho \leq (2(3/4)^{1-1/\alpha})^\alpha/2 = (3/2)^{\alpha-1}$. $\qquad\qquad\square$

5. Discussion

We discuss in this section practical issues that must be addressed to apply our results to production environments.

Heterogeneity of Servers. For the sake of simplicity, we assume in our model that all servers in a data center are identical. We believe this is reasonable, considering that modern data centers are usually built with homogeneous commodity hardware. Nevertheless, the proposed model and

31

derived results are also amenable to heterogeneous data center environments. In a heterogeneous data center, servers can be categorized into several groups with identical servers in each group. Then, different types of applications can be assigned to server groups according to their resource requirements. The VMA model presented here can be applied to the assignment problem of allocating tasks from the designated types of applications (especially CPU-intensive ones) to each group of servers. The approximation results we derive in this paper can be then combined with server-group assignment approximation bounds (out of the scope of this paper) for energy-efficient task assignment in real data centers, regardless of the homogeneity of servers.

Consolidation. Traditionally, consolidation has been understood as a bin packing problem [Mishra and Sahoo, 2011; Wang et al., 2011], where VMs are assigned to PMs attempting to minimize the number of active PMs. However, the results we derived in this paper, as well as the results in [Arjona Aroca et al., 2014a], show that such approach is not energy-efficient. Indeed, we showed that PM's should be loaded up to x^* to reduce energy consumption, even if this requires having more active PMs.

VM arrival and departure. When a new VM arrives to the system, or an assigned VM departs, adjustments to the assignment may improve energy efficiency. Given that the cost of VM migration is nowadays decreasing dramatically, our offline positive results can also be accommodated by reassigning VMs whenever the set of VM demands changes. Should the cost of migration be high to reassign after each VM arrival or departure, time could be divided in epochs buffering newly arrived VM demands until the beginning of the next epoch, when all (new and old) VMs would be reassigned (if necessary) running our offline approximation algorithm.

Multi-resource scheduling. This work focuses on CPU-intensive jobs (VMs) such as MapReduce-like tasks [Dean and Ghemawat, 2008] which are representative in production datacenters. As the CPU is generally the dominant energy consumer in a server, assigning VMs according to CPU workloads entails energy efficiency. However, there exist types of jobs demanding heavily other computational resources, such as memory and/or storage. Although these resources have limited impact on a server's energy consumption, VMs performance may be degraded if they become the bottleneck resource in the system. In this case, a joint optimization of multiple resources (out of the scope of this paper) is necessary for VMA.

Implementation on real systems. In [Jansen and Brenner, 2011; Mills et al., 2011b], the authors test multiple allocation algorithms which are already available in popular cloud platforms like OpenNebula [OpenNebula, 2013] or Eucalyptus [Eucalyptus, 2013]. Some examples of these allocation algorithms are Round Robin, First Fit, Least Full First or Next Fit among others. Including another allocation policy, such as our algorithms, in the cloud controllers of these and other platforms (e.g. Apache Mesos [Apache, 2014]) is feasible. Introducing our algorithms would make those platforms power efficient, providing power-aware allocation policies. This feature is not found on any of the algorithms mentioned in [Mills et al., 2011b] and only observed in the *Watts per Core* algorithm in [Jansen and Brenner, 2011]. We leave such integration for future work.

6. Conclusions

In this paper, we have studied a particular case of the generalized assignment problem with applications to Cloud Computing. We have considered the problem of assigning virtual machines (VMs) to physical machines (PMs) so that the power consumption is minimized, a problem that we call virtual machine assignment (VMA). In our theoretical analysis, we have shown that the decision version of (C, m)-VMA problem is strongly NP-complete. We have shown as well that the $(C, \cdot)$-VMA, $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA problems are strongly NP-hard. Hence, there is no FPTAS for these optimization problems. We have shown the existence of a PTAS that solves the $(\cdot, \cdot)$-VMA and $(\cdot, m)$-VMA offline problems. On the other hand, we have proved lower bounds on the approximation ratio of the $(C, \cdot)$-VMA and (C, m)-VMA problems. With respect to the online version of these problems, we have proved upper and lower bounds on the competitive ratio of the $(\cdot, \cdot)$-VMA, $(C, \cdot)$-VMA, $(\cdot, m)$-VMA, and (C, m)-VMA problems.

Our future work will consider the possibility that the load incurred by a VM changes over time or that the assignment of VMs to PMs is not final (and VMs can migrate, maybe at a cost). In fact, if the migration of VMs is available for free, our offline positive results can also be used in these new models, since an offline approximation algorithm can be run each time a load changes or a new VM arrives. Then, the VMs can be redistributed accordingly. Another future extension of the model will consider that the power consumption of a feasible solution of the VMA problem depends on several parameters simultaneously (e.g., memory space or communication bandwidth, in addition to processing load). Finally, as stated in Section 5, we plan to deploy our algorithm in a cloud

platform, probably OpenNebula, and compare the performance of our proposed algorithms against other state-of-the-art allocation algorithms such as the ones analyzed in [Jansen and Brenner, 2011; Mills et al., 2011b].

7. Acknowledgements

References

Alon, N., Azar, Y., Woeginger, G. J., Yadid, T., 1997. Approximation schemes for scheduling. In: Saks, M. E. (Ed.), SODA. ACM/SIAM, pp. 493–500.

Alon, N., Azar, Y., Woeginger, G. J., Yadid, T., 1998. Approximation schemes for scheduling on parallel machines. Journal of Scheduling 1 (1), 55–66.

Andrews, M., Anta, A. F., Zhang, L., Zhao, W., 2012. Routing for power minimization in the speed scaling model. IEEE/ACM Trans. Netw. 20 (1), 285–294.

Andrews, M., Antonakopoulos, S., Zhang, L., 2010. Minimum-cost network design with (dis)economies of scale. In: Proc. of 51-st Annual IEEE Symposium on Foundations of Computer Science. pp. 585–592.

Antoniadis, A., Im, S., Krishnaswamy, R., Moseley, B., Nagarajan, V., Pruhs, K., Stein, C., 2014. Hallucination helps: Energy efficient circuit routing. In: [Erlebach and Persiano, 2013].

Apache, 2014. Apache mesos. `http://http://mesos.apache.org/`, accessed December 11th, 2014.

Arjona Aroca, J., Chatzipapas, A., Fernández Anta, A., Mancuso, V., 2014a. A measurement-based analysis of the energy consumption of data center servers. In: e-Energy. ACM.

Arjona Aroca, J., Fernández Anta, A., Mosteiro, M. A., Thraves, C., Wang, L., 2014b. Power-Efficient Assignment of Virtual Machines to Physical Machines. Lecture Notes in Computer Science. Springer International Publishing.

URL http://dx.doi.org/10.1007/978-3-319-13464-2_6

Bansal, N., Chan, H.-L., Pruhs, K., 2009. Speed scaling with an arbitrary power function. In: Proc. of 20-th Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 693–701.

Bansal, N., Gupta, A., Krishnaswamy, R., Nagarajan, V., Pruhs, K., Stein, C., 2012. Multicast routing for energy minimization using speed scaling. In: MedAlg. pp. 37–51.

Bellur, U., Rao, C. S., SD, M. K., 2010. Optimal placement algorithms for virtual machines, arXiv:1011.5064 (http://arxiv.org/abs/1011.5064).

Beloglazov, A., Abawajy, J., Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Generation Computer Systems 28 (5), 755 – 768, special Section: Energy efficiency in large-scale distributed systems.

Botero, J. F., Hesselbach, X., Duelli, M., Schlosser, D., Fischer, A., de Meer, H., 2012. Energy efficient virtual network embedding. IEEE Communications Letters 16 (5), 756–759.

Cardosa, M., Singh, A., Pucha, H., Chandra, A., 2011. Exploiting spatio-temporal tradeoffs for energy-aware mapreduce in the cloud. In: Cloud Computing (CLOUD), 2011 IEEE International Conference on. pp. 251 –258.

Chakrabarty, D., Chekuri, C., Khanna, S., Korula, N., 2011. Approximability of capacitated network design. In: IPCO. pp. 78–91.

Chandra, A. K., Wong, C. K., 1975. Worst-case analysis of a placement algorithm related to storage allocation. SIAM J. Comput. 4 (3), 249–263.

Chen, S.-C., Lee, C.-C., Chang, H.-Y., Lai, K.-C., Li, K.-C., Rong, C., 2011. Energy-aware task consolidation technique for cloud computing. In: Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science. pp. 115–121.

Citrix, 2012. Citrix. http://www.citrix.com, accessed August 27, 2012.

Cody, R. A., Jr., E. G. C., 1976. Record allocation for minimizing expected retrieval costs on drum-like storage devices. J. ACM 23 (1), 103–115.

Dean, J., Ghemawat, S., 2008. Mapreduce: simplified data processing on large clusters. Commun. ACM 51 (1), 107–113.

EC2, A., 2012. Amazon web services. `http://aws.amazon.com`, accessed August 27, 2012.

Epstein, L., Sgall, J., 2004. Approximation schemes for scheduling on uniformly related and identical parallel machines. Algorithmica 39 (1), 43–57.

Erlebach, T., Persiano, G. (Eds.), 2013. Approximation and Online Algorithms - 10th International Workshop, WAOA 2012, Ljubljana, Slovenia, September 13-14, 2012, Revised Selected Papers. Vol. 7846 of Lecture Notes in Computer Science. Springer.

Espadas, J., Molina, A., Jiménez, G., Molina, M., Ramírez, R., Concha, D., 2013. A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures. Future Generation Computer Systems 29 (1), 273 – 286, including Special section: AIRCC-NetCoM 2009 and Special section: Clouds and Service-Oriented Architectures.

Eucalyptus, 2013. Eucalyptus. `http://www.eucalyptus.com/`, accessed January 20th, 2013.

Fernandez de la Vega, W., Lueker, G., 1981. Bin packing can be solved within $1 + \epsilon$ in linear time. Combinatorica 1 (4), 349–355.
URL `http://dx.doi.org/10.1007/BF02579456`

Garey, M. R., Johnson, D. S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York, NY, USA.

Gupta, A., Krishnaswamy, R., Pruhs, K., 2012. Online primal-dual for non-linear optimization with applications to speed scaling. In: [Erlebach and Persiano, 2013], pp. 173–186.

Im, S., Moseley, B., Pruhs, K., 2012. Online scheduling with general cost functions. In: Proc. of 23-rd Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 1254–1265.

Jansen, R., Brenner, P., 2011. Energy efficient virtual machine allocation in the cloud. In: Green Computing Conference and Workshops (IGCC), 2011 International. pp. 1–8.

Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., Jiang, G., 2009. Power and performance management of virtualized computing environments via lookahead control. Cluster Computing 12 (1), 1–15.

Liu, N., Dong, Z., Rojas-Cessa, R., 2012. Task and server assignment for reduction of energy consumption in datacenters. In: Proceedings of the IEEE 11-th International Symposium on Network Computing and Applications. pp. 171–174.

Machida, F., Kawato, M., Maeno, Y., 2010. Redundant virtual machine placement for fault-tolerant consolidated server clusters. In: Network Operations and Management Symposium (NOMS), 2010 IEEE. pp. 32 –39.

Mark, C., Niyato, D., Chen-Khong, T., 2011. Evolutionary optimal virtual machine placement and demand forecaster for cloud computing. In: Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on. pp. 348 –355.

Mills, K., Filliben, J., Dabrowski, C., 2011a. Comparing vm-placement algorithms for on-demand clouds. In: Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science. pp. 91–98.

Mills, K., Filliben, J., Dabrowski, C., 2011b. Comparing vm-placement algorithms for on-demand clouds. In: Proceedings of the 2011 IEEE Third International Conference on Cloud Computing Technology and Science. CLOUDCOM '11. IEEE Computer Society, pp. 91–98.

Mishra, M., Sahoo, A., 2011. On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach. In: Cloud Computing (CLOUD), 2011 IEEE International Conference on. pp. 275 –282.

Nathuji, R., Schwan, K., 2007. Virtualpower: coordinated power management in virtualized enterprise systems. In: SOSP. pp. 265–278.

Nguyen Van, H., Dang Tran, F., Menaud, J.-M., 2009. Autonomic virtual resource management for service hosting platforms. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. CLOUD '09. IEEE Computer Society, pp. 1–8.

OpenNebula, 2013. Opennebula. `http://opennebula.org/`, accessed January 20th, 2013.

Pop, F., Potop-Butucaru, M. (Eds.), January 2015. Adaptive Resource Management and Scheduling for Cloud Computing. Vol. 8907 of Lecture Notes in Computer Science (LNCS) / Theoretical Computer Science and General Issues. Springer.

Rackspace, 2012. Rackspace. `http://www.rackspace.com`, accessed August 27, 2012.

Srikantaiah, S., Kansal, A., Zhao, F., 2008. Energy aware consolidation for cloud computing. In: Proceedings of the 2008 conference on Power aware computing and systems. HotPower'08. USENIX Association, pp. 10–10.

Van den Bossche, R., Vanmechelen, K., Broeckhove, J., 2011. Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds. In: Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science. pp. 320–327.

Vazirani, V. V., Mar. 2004. Approximation Algorithms. Springer.
URL `http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20\&path=ASIN/3540653678`

Viswanathan, H., Lee, E., Rodero, I., Pompili, D., Parashar, M., Gamell, M., 2011. Energy-aware application-centric vm allocation for hpc workloads. In: Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), 2011 IEEE International Symposium on. pp. 890 –897.

Wang, M., Meng, X., Zhang, L., 2011. Consolidating virtual machines with dynamic bandwidth demand in data centers. In: IEEE INFOCOM. pp. 71–75.

Xu, J., Fortes, J., 2011. A multi-objective approach to virtual machine management in datacenters. In: Proceedings of the 8th ACM international conference on Autonomic computing. ICAC '11. ACM, pp. 225–234.