Selecting services in the cloud: a decision support methodology focused on infrastructure-as-a-service context

Cássio L. M. Belusso, Federal University of Fronteira Sul, Av. Jacob Reinaldo Haupenthal, s/n, Cerro Largo, 97900-000, Brazil

Sandro Sawicki, UNIJUI University, Rua do Comércio, 3000, Ijuí, 98700-000, Brazil

Vitor Basto-Fernandes, University Institute of Lisbon (ISTAR-IUL), Av. das Forças Armadas, 376, 1649-026, Lisbon, Portugal

Rafael Z. Frantz, UNIJUI University, Rua do Comércio, 3000, Ijuí, 98700-000, Brazil

Fabricia Roos-Frantz, UNIJUI University, Rua do Comércio, 3000, Ijuí, 98700-000, Brazil

Keywords: Cloud computing; Multicriteria decision-making; Analytic hierarchy process; Cloud migration; Pareto dominance

Abstract
Growing demand for reduced local hardware infrastructure is driving the adoption of Cloud Computing. In the Infrastructure-as-a-Service model, service providers offer virtualized computational resources in the form of virtual machine instances. The existence of a large variety of providers and instances makes the decision- making process a difficult task for users, especially as factors such as the datacenter location - where the virtual machine is hosted - have a direct influence on the price of instances. The same instance may present price differences when hosted in dif- ferent geographically distributed datacenters and, because of that, the datacenter location needs to be taken into account through the decision-making process. Given this problem, we propose the D-AHP, a methodology to aid decision-making based on Pareto Dominance and Analytic Hierarchy Process (AHP). In the D-AHP, the dominance concept is applied to reduce the number of instances to be compared; the instances selection is based on a set of objectives, while AHP ranks the selected ones from a set of criteria and sub-criteria, among them the datacenter location. The results from case studies show that differences may arise in the results, regarding which instance is more suitable for the user, when considering the datacenter loca- tion as a criterion to choose an instance. This fact highlights the need to consider this factor during the process of migrating applications to the Cloud. In addition, Pareto Dominance applied early over the set of total instances has proved to be efficient, once it significantly reduces the number of instances to be compared and ordered by the AHP by excluding instances with less computational resources and higher cost in the decision-making process, mainly for larger application workloads.

1 Introduction
Cloud Computing has emerged as one of the most significant advancements in the field of Information Technology (IT) because of its advantages over local hardware infrastructures for aspects such as agility, elasticity [1], flexibility, and cost. Because of these attractive features, the International Data Corporation (IDC) estimates that spending on public cloud services is expected to reach US$ 370 billion in 2022 [2].

In Cloud Computing, providers offer computing services on a pay-per-use basis [3], providing significant savings in resources related to investment, management, and maintenance of local infrastructure. Software-as-a-Service (SaaS), Platform- as-a-Service (PaaS), and Infrastructure-as-a-Service (IaaS) are among the services offered by the providers [4].

Cloud Computing has been attracting interest from the scientific community. One of the identified gaps which still persists is lack of transparency on the part of providers regarding the pricing of Virtual Machine (VM) instances, and the variety (and constraints) of corresponding services when it comes to using these models in the decision-making process [5]. Thus, the decision to migrate is still considered complex due to the immaturity and dynamics of this environment. Even so, in the business field, migration is a strategic decision that can improve performance, pro- ductivity and growth, and increase competitiveness [6, 7].

In the IaaS model, the prices of VM instances are based on a set of numerical var- iables (such as the CPU, memory, and storage amounts) and non-numeric variables (such as the VM operating system (OS) and the geographic datacenter location).

To gain a better understanding of the problems associated with selection of ser- vices, researchers have followed different approaches. The studies of Li et al. [8], Kihal et al. [9], Menzel and Ranjan [10], Murthy et al. [11], Menzel et al. [12], Emeras et al. [13], López-Pires and Barán [14], Mitropoulou et al. [15], Al-Faifi et al. [16], Nagarajan and Thirunavukarasu [17], and Chauhan et al. [18] used CPU, memory, and storage resources simultaneously as decision variables. However, they did not present mechanisms to reduce the dimensionality of the problem; in addi- tion, many of these studies do not carry out a multi-provider approach, which may influence choice of services. Studies by Li et al. [8], Yao et al. [19], Malekimajd et al. [20], Souidi et al. [21], Menzel et al. [12], and Ziafat and Babamir [22] have considered the datacenter location as a decision variable, although they analyzed it in relation to the performance of the network. Mitropoulou et al. [15] have analyzed it in relation to cost, presenting regression models with low significance.

Datacenter location is one of the factors that most affects the price of instances. In addition, the lack of transparency regarding the variables which affect price dif- ferences of instances from different datacenters, plus the wide variety of geographic locations distributed around the world, turn the decision-making process difficult.

As a result, a user who wants to change datacenter for some reason without increasing the cost (for example, to improve latency) may be forced to select another instance with fewer computational resources in order to maintain a financial bal- ance. In such cases, the ideal scenario would be to select an instance that does not have significant price variations in different datacenter locations.

Some important issues may arise while selecting a VM instance. When pri- oritizing cost, users tend to select lower-priced instances, which may compro- mise the performance of the application to be migrated, since the instance may not have enough computational resources to guarantee good performance of the application. When prioritizing computational resources, users tend to select instances with large capacity in terms of resources such as the CPU, memory, and storage. However, if the application does not need all the capacity available, the VM tends to become idle, making the migration process expensive. In both cases, in order to make the best decision and optimize the resources involved in the decision-making process, it is important to be aware of the requirements of the application to be migrated, in order for the cost-benefit ratio to be opti- mized.

Therefore, users must have a methodology that facilitates choice of a VM instance by

reducing the number of available instances and which allows selec- tion based on their preferences.
The lack of tool support to automate migration tasks is highlighted by Jam- shidi et al. [23], whose work consists in the systematic analysis of studies about the migration of legacy applications from local infrastructures to the Cloud. In addition, the authors identify a need for both architectural adaptation and self-adaptive Cloud- enabled systems.
In this sense, how can users be assisted during the decision-making process when choosing a provider and a VM instance when they are migrating their applications to an IaaS in the Cloud? Are the computational resources of the VM sufficient for the execution of such applications, considering the lowest price? Furthermore, how can the number of instances be reduced so that users have a smaller set of VM instances to compare, thereby reducing the resources involved in the decision-making process?
In order to assist with this process, we present the D-AHP, a methodology to help the decision-making of a VM instance be more suitable for migration of applica- tions, using the Pareto Dominance concept and the multicriteria approach of the Analytic Hierarchy Process (AHP). For this, we define decision criteria as the price and amount of computational resources that comprise the VM instance. In addition, as a differential, we define different datacenter locations as sub-criteria due to the large influence of this variable on the price criterion.
In summary, we can cite the following contributions:
– To propose an easy-to-understand decision-making methodology, allowing an analysis based on the importance level of the decision criteria for the selection of VM instances, assuming as a premise that the prioritization of one criterion in relation to others tends to accommodate the needs of users better;
– To analyze the influence of datacenter location on the price of VM instances imposed by providers, proving its importance in decision-making processes;
– To reduce the dimensionality of the VM instances selection problem, by reduc- ing the number of comparable instances, thus providing greater agility to the decision-making process.
The remainder of this paper is organized as follows: Sect. 2 highlights related studies; in Sect. 3, the problem modeling is done; Sect. 4 presents details of the D-AHP methodology; Sect. 5 presents different case studies for the validation of the proposed methodology; Sect. 6 presents the conclusions.

2 Related work
A great number of researchers have focused their efforts on solving problems that involve IaaS selection to assist in the migration of applications to the Cloud by analyzing performance-related variables such as CPU, memory, storage, and cost, from the point of view of both the provider and user [24]. For this, they use dif- ferent approaches; among them, the use of heuristics [25] and the use of classifi- cation and ordering methods.
In this section, the related and described studies were divided into two groups, both in the IaaS context. In the first one, we present studies that cover the selec- tion of services and modeling of prices; in the second group, we present stud- ies that use optimization techniques for the selection of services. Based on the related works shown in Tables 1 and 2, we detail only of those that addresses the datacenter location in its proposals.

Generally, studies that deal with IaaS selection processes in the Cloud admit, as the main hypothesis, the reduction of costs, directing more attention to the analysis of the policies adopted by the providers to define the prices of the VM instances. Thus, researchers seek to detect the variables that most influence prices, among which are the computational resources and the datacenter location.

In Table 1, we present a list of related studies and the variables most com- monly used by researchers within the context of the selection of services and price modeling.

For Al-Roomi et al. [46] and Mazrekaj et al. [57], despite the attempt to make the pricing policy practiced by providers more transparent through the search for an exact formula, the definition of these prices is made considering several fac- tors, including commercial ones, which makes this task more complex.

Table 1 Variables analyzed in related work to the context of selecting IaaS in the Cloud

Variable	Reference (by Year)	
	2010–2015	2016–2020
CPU	[8–12, 26–29]	[13–18, 30–34]
Memory	[8–12, 26–28, 35]	[13–15, 17, 18, 30–33, 36, 37]
Storage	[8–12, 29, 38–40]	[13–15, 17, 18, 32, 41, 42]
Datacenter Location	[8, 10, 12, 19, 21]	[15, 22, 43]
Cost	[8–12, 26–29, 35, 38, 40, 44–49]	[13–17, 30–33, 36, 37, 41, 42, 50–62]
Others	[8, 10, 11, 21, 28, 29, 35, 39, 40, 44, 46, 47, 49, 63]	[16–18, 30, 36, 37, 41, 42, 57–59, 61, 64]

Table 2 Optimization methods used in related work for the context of selecting IaaS in the Cloud

Method	Reference (by Year)	
	2010–2015	2016–2020
AHP/Fuzzy AHP	[10, 12, 63, 67–71]	[18, 30, 52, 60, 72–76]
TOPSIS/Fuzzy TOPSIS	[28, 63, 77]	[33, 42, 52, 74, 75, 78]
ELECTRE	[63, 79]	
PROMETHEE	[49, 63]	[72]
Genetic Algorithm	[10, 12, 80–82]	[55, 56, 83–90]
Particle Swarm Optimization	[82, 91]	[83, 85, 92]
Ant Colony Optimization	[81, 93]	[94]
Simulated Annealing		[56, 61]
Others	[49, 63, 68, 95–97]	[16–18, 30–32, 42, 54, 56, 58, 75, 76, 84, 86, 87, 89, 90, 98–102]

Thus, in the case of a problem composed of several variables that involve the selection of IaaS in the Cloud, optimization methods are presented as a good alter- native to assist in the service selection process.

Due to the wide applicability of optimization methods in problems of this nature, Alabool et al. [65] carried out a systematic review of studies that propose the use of Multicriteria Decision-Making (MCDM), in order to develop Cloud Service Evalua- tion Methods (CSEMs). The authors employed the Evaluation Theory to detect defi- ciencies in each proposal and created an information base so that researchers can better develop their CSEMs.
Hosseinzadeh et al. [66] presented an overview on a set of articles suggesting the use of MCDM methods to develop schemes for services selection in the Cloud. In order to evaluate the proposed method, the authors identified the optimization method, the Quality-of-Service (QoS) criteria, and the set of data and environments utilized.
In Table 2, we present a list of related studies and the optimization method used by each one. It can be noticed that, in multiobjective optimization, the most used method is the Genetic Algorithm, while in multicriteria optimization the most used method is the AHP and its variants.
In relation to the study that addresses the datacenter location in its proposals, Mitropoulou et al. [15] propose the construction of a price index based on a hedonic method of pricing. By using regression models, the authors analyzed a set of factors that affect the final price of VM instances, among them the datacenter location. The authors collected data from providers on a specific platform and analyzed in which regions they offer services, grouping them by continent. However, the models pre- sented low significance.
Menzel and Ranjan [10] and Menzel et al. [12] presented CloudGenius, a frame- work based on the principles of AHP and the Genetic Algorithm to assist in the migration of Web applications. CloudGenius addresses the decision-making process based on three main goals: lower price, better latency, and better QoS. However, the authors evaluate the effects of datacenter location only on network performance, regardless of costs.
Souidi et al. [21] used as a hypothesis of the selection problem the option of data- center location based on the location of the user, aiming at a better performance of the network. A similar approach is adopted by Li et al. [8], who assumed that the lower the distance between the datacenter location and the location of the user, the lower the latency of the network. Network latency in geographically distributed datacenters is also analyzed by Yao et al. [19] and Malekimajd et al. [20]. However, in none of these studies, the influence of the datacenter location on the cost of ser- vices in the Cloud is verified.
Ziafat and Babamir [22, 43] used different multiobjective optimization algorithms for the selection of datacenters considering qualitative aspects such as the distance between datacenter and user, indices of reliability and availability, response time, and lower cost. However, when considering a large number of conflicting objectives, the selection of a datacenter that satisfies all the objectives, despite being considered optimal by the optimization algorithm, tends not to satisfy user needs effectively, since it is not possible to prioritize an objective in relation to another.
In relation to the datacenter location, Marks and Lozano [103] highlight some important aspects to be considered, namely cost, since the same instance which is hosted in datacenters from different locations may present differences in its final price; data transmission, once possible delays may occur (due to the distance between the datacenter and the user), therefore compromising a quality service; and confidentiality of information, as some countries have specific laws regarding data hosting policy within its boundaries.
In Table 3, we summarize a comparison between our proposed approach and other state-of-the-art approaches that analyze datacenter location.

Given this scenario, our proposal differs from those presented because, in addi- tion to analyzing all the resources that compose the instances, simultaneously, it analyzes the datacenter location in relation to the cost aspect. In a complementary way, when comparing our proposal with those that analyze the influence of the data- center location on the cost, the difference is that it allows the user to prioritize one objective over the others, making it adaptable for the user to define their preferences based on the requirements of their applications. In this way, we are not aware of any similar study in terms of adopted methodology, decision-making criteria, datacenter location analysis in the price of VMs, and combination of the optimization tech- niques used related to the selection of IaaS in the Cloud.

3 Problem modeling

When decision-makers (DMs) planning, for strategic reasons, the migration of applications from local infrastructures to an IaaS in the Cloud (for example, due to a cost reduction, market trends or a large volume of data), they should initially know whether it is possible or not. In addition, they need to decide whether to total or par- tial migration, which implies re-writing all the application or parts of it.

Table 3 Comparison between our proposed approach and other state-of-the-art approaches that analyze datacenter location

Reference	Technique(s)	Objective(s)	Limitation(s)	Tool(s)
[8]	CloudCmp	Speed of CPU, memory, and disk I/O, scaling latency, storage service response time, time to reach consistency, network latency, and available bandwidth	Do not select instances, just provider; do not analyze the influence of datacenter location on the cost	Pre-existing tools specific to each metric
[10, 12]	AHP method; Genetic Algorithm	Cheapest service, best latency, best quality (performance, uptime, popularity)	For web applications only; do not analyze the influence of datacenter location on the cost	Aotearoa AHP; Mahout framework for hadoop
[19]	Two-dimensional knapsack algorithm	Network latency	Only network factors are considered.	Comparison with Random algorithm and Greedy algorithm
[21]	SCPS and ACPS algorithms	Network performance, VM cost, availability, trustworthiness, VM performance	Do not select instances, just provider; do not analyze the influence of datacenter location on the cost	Key Performance Indicators (KPI); Page Hinkley; VM instances sub-scribed
[15]	Hedonic price index	CPU, memory, storage, OS, Transfer_OUT, subscription	Continent-based approach, not by datacenter individually	Cloudorado
[22, 43]	NSGAII_Cluster algorithm; k-means algorithm	Cost, reliability, availability, response time	Do select datacenters, not instances and providers	NetBeans, CloudSim
[20]	TreeOpt, NetOpt, and TreeOptExt algorithms	Network latency	Only network latency is considered	Comparison with algorithm Min-DiameterGraph and algorithm for random selection
Our	Pareto Dominance; AHP method	Cost and amount of computational resources (CPU, memory, and stor-age) and cost analysis (datacenter location)	Data are not collected in real-time	MATLAB, Cloudorado

Regardless of the type of migration, DMs need to select an IaaS provider from a set of providers $P = \{p_1, p_2, \ldots, p_t\}$ capable of offering a better QoS at a low cost. These providers will present an extensive set of instances $I = \{i_1, i_2, \ldots, i_n\}$ which are composed of various computational resources R_n , such that:

$$C_T = C_R + C_L + C_{OS} \quad (2)$$

The same instance hosted in different datacenter locations can have significantly different prices. For example, data obtained in July 2018 showed that Amazon's instance r3.2xlarge (8 vCPU, 61 GB of memory, and 160 GB of SSD storage), hosted in Brazil, had been priced at US$ 1.3990/hour. The same instance, hosted

in the USA, had been priced at US$ 0.6650/hour. In comparison, the instance r3.4xlarge (16 vCPU, 122 GB of memory, and 320 GB of SSD storage) had been priced at US$ 1.3300/hour, i.e., double the resources for half the price [104].
This price difference cannot be justified just by changing the datacenter location. Additional factors contribute to such difference, some of them related to the VM configurations, as vCPU cores, I/O rates, and older hardware. However, this information is not clear enough and many times difficult to be accessed by users with low technical knowledge. Thus, it is common that the first thing the user seeks is the number of computational resources and the price of the VM - both easy and quick information to access.
Therefore, services are generally selected which offer greater amounts of computational resources at the lowest price, according to the requirements of the applications in question, and different priorities may be assigned to the decision criteria. Thus, in this case, we assume that the best option is to choose the provider $p_T \in P$ and the instance $i_n \in I$ from which one can obtain the best cost-benefit relation, characterized by maximizing instance resources R_n and minimizing the final cost C_T, such that:

$$\max_{i_n \in I} R_n \tag{3}$$

$$\min_{i_n \in I} C_T \tag{4}$$

4 D-AHP methodology
In this section, we present the D-AHP, a decision-making methodology to support processes for migrating applications to computational infrastructures in Cloud environments.
We propose use of the D-AHP methodology to solve the following problem:
Select a VM instance $i_n \in I$ associated with an IaaS provider $p_t \in P$, in such a way as to obtain the maximum amount of computational resources R_n at the lowest price C_T, based on CPU, memory, and storage amounts, and the prices of the instances in different datacenter locations L.
To do this, the D-AHP analyzes a set of variables arranged in two distinct sets: the set of numerical variables $V = \{v_1, v_2, \dots, v_x\}$ and the set of non-numeric variables $\bar{V} = \{\bar{v}_1, \bar{v}_2, \dots, \bar{v}_y\}$, as shown in Tables 4 and 5.
The D-AHP basically consists of four main steps included in three major phases, as shown in Fig. 1.
The first step is to pre-select trusted providers that have a good range of services, comply with Service Level Agreement (SLA) terms and are able to quickly adapt to the characteristic dynamics of the Cloud environment.

Table 4 Set of numerical variables V

Variable	Description	Metric
Q_{MEM}	Memory	GB
Q_{CPU}	CPU	Number of vCPUs (cores)
Q_{STO}	Storage	GB
C_T	Cost	US Dollars

Table 5 Set of non-numerical variables $\overline{V}$

Variable	Description	Alternatives
L	Datacenter location	Country 1,..., Country ξ
O_S	Operating system	Linux, Windows
S_A	Storage system	HDD, SSD
I_T	Instance type	On-demand
T_N	Deployment model	Public Cloud

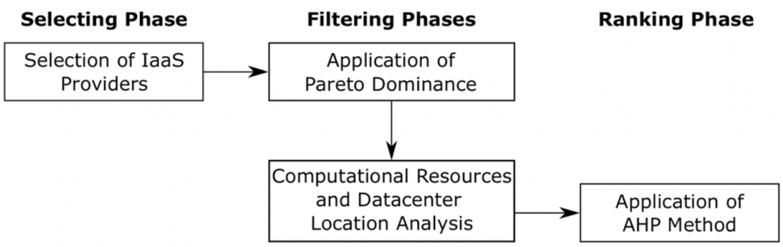

Fig. 1 Major phases of D-AHP

The second step is to apply Pareto Dominance [105] to a set of computational resources and its price in order to reduce the number of instances. As a result, only the non-dominated instances are selected for the next step.

In the third step, the non-dominated instances are analyzed in relation to their availability regarding datacenter locations and the computational requirements of the application to be migrated. The instances resulting from this new filtering give rise to the set of selectable instances, arranged at the last level of the hierarchical structure present in the final step of the D-AHP.

The fourth and final step of the D-AHP is to use the AHP method [106] to obtain a final classification of the selectable instances. In this process, they are compared to one another from the perspective of a set of decision criteria and sub-criteria which, as in the second step, are related to the amount of computational resources and price.

The D-AHP is adaptable to any OS or storage system. In addition, it is possible to use it by considering a set of free-choice datacenter locations, or even only among availability zones within the same region.

Moreover, the D-AHP considers that the application workload to be migrated from local infrastructures is constant. If we consider a dynamic application workload, the QoS values can be significantly changed, and problems of over-provisioning or under-provisioning of computational resources can be detected [107].

In the following subsections, we detail the steps in the D-AHP and specify the processes performed internally in each step. Also, we represent these steps and processes by a flowchart, as shown in Fig. 2.

4.1 Step 1: selection of IaaS providers

In this first step, we seek to select a subset of providers $P^{+} \in P = \{p_1, p_2, \ldots, p_\tau\}$ so that the next steps in the D-AHP can be constructed.

In the D-AHP proposal, users define the set of IaaS providers using inclusion or exclusion criteria, according to their preferences [108]. In this study, providers are selected using two approaches: the first is by definition of a set of services offered to the DMs when migrating their applications to the Cloud; the second is reference to Gartner's Magic Quadrant for updated IaaS providers [109].

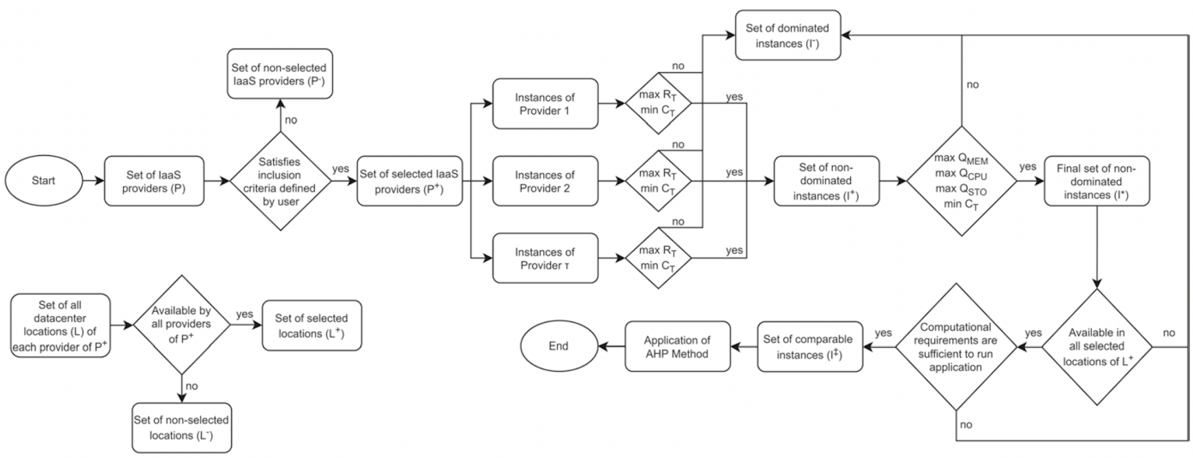

Fig. 2 Decision-making process of D-AHP

The services adopted as criteria for choosing the set of selected providers P^+ to be analyzed (15 providers) were as follows: Any Location (by continent), Hourly Pay-As-You-Go, Auto-Scaling, No Limit Transfer, Support 24x7, Load Balancing, Firewall, Operating System, GPU Instances, and SSD Storage [110].

The annual Magic Quadrant developed by Gartner uses aspects such as Ability to Execute and Completeness of Vision to classify IaaS providers for a given period of market observation. Among the ranking groups, the Leader providers are technologically more advanced. They are points of reference within the industry and they dictate the rules within the segment by having a better view of the market and the ability to carry forward the results of their research.

Thus, we aim to verify, from the providers that are members of the Leaders' quadrant, which ones offer the complete set of defined services. Joint analysis of both factors will identify providers belonging to the set of selected providers P+ and the set of unselected providers P^-.

4.2 Step 2: applying pareto dominance

According to the Pareto Dominance, if X^* is the set of feasible solutions to a minimization problem and if $x, x^* \in X^*$, then solution x dominates x^* if, and only if, f(x) is better than $f(x^*)$ in at least one objective, such that $f_i(x) < f_i(x^*)$, and it is not worse in any other, such that $f_j(x) \leq f_j(x^*)$, for i, j = 1, 2,..., k and $i \neq j$, where f is the objective function. If both have the same level of dominance, then $f(x) \nleq f(x^*)$ and $f(x^*) \nleq f(x)$, and x is indifferent to x^* [105].

Thus, for a solution to be non-dominated, there must be no other solution within the search space better than it, whenever all objectives are simultaneously considered. When applying Pareto Dominance in a set, we look for the best solutions belonging to it, with the best performance in relation to multiple objectives, which can be maximization or minimization.

In the D-AHP proposal, Pareto Dominance is applied in order to reduce the number of instances provided by the providers of set P^+. As a direct consequence, the number of pairwise comparisons is significantly reduced, which is the basis of the AHP method (applied in the last step of the D-AHP).

In the D-AHP, the dominance relationship is applied initially intra-provider, that is, in instances of the same provider. In our analysis, we only used On-demand instances in the Public Cloud model.

From Eq. 1, it is known that $R_n = (Q_{MEM}, Q_{CPU}, Q_{STO})$. Therefore, the maximization of R_n is conditioned to the maximization of its three components. Thus, the dominance relationship is applied considering four objectives and no prioritization among them, according to Eqs. 5–8.

$$\max_{i_n \in I} Q_{\mathrm{MEM}} \tag{5}$$

$$\max_{i_n \in I} Q_{\mathrm{CPU}} \tag{6}$$

$$\max_{i_n \in I} Q_{\mathrm{STO}} \tag{7}$$

$$\min_{i_n \in I} C_T \tag{8}$$

The dominance relationship is applied over the set of instances as follows: for example, according to the data obtained in July 2018, the provider Azure was offering instance H8 at a price of US$ 0.796/hour; the same provider was offering instance L8 at a price of US$ 0.624/hour [111]. According to Table 6, pairwise comparison of H8 and L8 showed that the amount of computational resources of L8 were bigger than (Memory and Storage) or equal to (vCPU) those of H8. Besides this, L8 had been priced lower than H8. Therefore, L8 dominates H8; i.e., L8 is non-dominated, and H8 is dominated.

After this process is performed for all providers of set P+ selected in Step 1, the non-dominated and dominated instances of each are stored in the non-dominated and dominated instances sets, denoted by I^+ and I^-, respectively. Cases where there

are indifferent instances, both are also added to set I^+. An instance, when included in the set I^-, is automatically eliminated from the next steps of the process, as it has already been dominated by some other instance in relation to all objectives.
A new dominance relationship is applied in an inter-provider way over the instances of set I^+, which is composed of non-dominated instances from all providers of set P^+. Dominated instances are stored in set I^-, together with the dominated ones resulting from the first dominance relationship, while the non-dominated ones are stored in the final set of non-dominated instances I^*, which must then pass through a new filtering process in the third step of the D-AHP. Until this step, the computational demand of the application has not been analyzed.

4.3 Step 3: instance filtering

In this step, two constraints are established, applicable to the set of non-dominated instances I^*.
The first constraint refers to the availability locations of the instances, which is necessary because of the multi-provider approach of the D-AHP. We assume that instances can only have their prices compared if they are hosted in datacenters whose locations L are common among the providers of set P^+. Thus, the selected and unselected locations are stored in sets L^+ and L^-, respectively.

Table 6 Pairwise comparison between instances H8 and L8 [111]

Instance	H8	L8
vCPU	8	8
Memory (GB)	56	64
Storage (GB)	1000	1388
Price (US$/hour)	0.796	0.624

When this constraint is applied, instances that are not available in all common datacenter locations among providers of set P^+ are omitted from selection for the next step; when this constraint is not applied, DMs can be prevented from expanding their searches for more attractive prices in other regions, invalidating one of the objectives of the D-AHP, which is the search for the lowest price.
However, the adaptability of the D-AHP in relation to the number of datacenter locations should be noted. The D-AHP allows DMs to increase or decrease the number of datacenter locations when they want to analyze the prices of instances. In cases where the DM already has a defined provider, it is possible to apply the D-AHP only to this provider or even just to availability zones within the same region.
The second constraint refers to the demand for computational resources consumed by the workload of the application to be migrated. Instances whose computational resources are lower than the demand of the application are not selected because it is assumed that there will not be enough resources to execute the application if one of these instances is selected during the process.
Application of both constraints creates set $I^{\ddagger}$, which is composed of the selectable instances to be arranged at the last level of the hierarchical structure of the final step

of the D-AHP, whose number of instances tends to be smaller than the number of instances of set I^* . This provides a smaller number of pairwise comparisons, and, as a result, there is a lower possibility of inconsistencies occurring during the DMs' judgment.

4.4 Step 4: elements of the AHP method

The AHP method allows to structure a problem in the form of a hierarchy of criteria, which has at least three levels: at the top, the main objective O of the problem; in the middle, the set of decision criteria $C = \{C_j \mid j = 1, 2,\ldots,m\}$ that define the alternatives; and at the bottom, the set of competing alternatives $A = \{A_i \mid i = 1, 2,\ldots, n\}$.

The basis of the AHP consists of a pairwise comparison between the elements of each hierarchical level. For such, the Saaty scale is used, whose values vary from 1 to 9 to represent the importance level between two criteria, in which 1 means the equal importance level; 3, 5, 7, and 9 mean the moderate, strong, very strong, and extreme importance level of one criterion over another, respectively; and 1/3, 1/5, 1/7, and 1/9 represent reciprocal importance levels [112].

The elements resulting from the pairwise comparison are arranged in a comparison matrix M, whose elements represent the dominance level between two criteria. Based on the elements of M, the weight vector of the criteria j can be obtained through the geometric mean method [106].

To obtain the final classification of the set of alternatives, an aggregation process is carried out, that is, the global valuation of A_i in relation to the main objective following the weighted sum model.

Let $V(A_i)$ be the global value of A_i in relation to O, w_j the preference level (weight) of the jth criterion in relation to O, and D the decision matrix whose elements x_{ij} represent the preference level of Ai in relation to the criterion Cj . Therefore, V(Ai) is calculated by Eq. 9.

$$V(A_i) = \sum_{j=1}^{m} x_{ij} \cdot \omega_j^T; \quad i = 1, 2, \ldots, n \tag{9}$$

where w_j^T is the transpose of w_j.

The alternatives are classified according to their respective global values. The best alternative A_{best} is the one with the highest global value, according to Eq. 10.

$$A_{best} = \max[V(A_i)]; \quad i = 1, 2, \ldots, n \tag{10}$$

The verification of possible inconsistencies of judgments from contradictory comparisons can be calculated by the Consistency Index (CI) and the Consistency Ratio (CR), according to Eqs. 11 and 12. If CI, CR < 0.1 , then there is consistency in the judgments; if not consistent, judgments must be redone.

$$CI = \frac{\lambda_{max} - m}{m - 1} \quad (11)$$

$$CR = \frac{CI}{RI} \quad (12)$$

where λ_{max} is largest eigenvalue of the matrix M, m is the number of criteria and RI is the Random Consistency Index, whose values are shown in Table 7.
The hierarchy proposed in the D-AHP methodology is represented in Fig. 3. It is composed of a main objective, two criteria, ξ sub-criteria, and a set of n instances as decision alternatives whose number may vary depending on the result of the filtering in Steps 2 and 3.
Below, we describe each of the elements of the hierarchy of the D-AHP shown in Fig. 3. These include the following:

– **Objective: Select VM Instance** - The aim is to select an instance with enough computational resources for execution of the application to be migrated, at the lowest price, optimizing the cost-benefit ratio;
– **Criterion 1: Computational Resources** - This refers to the amounts of computational resources of VM instances, specifically CPU, memory, and storage;
– **Criterion 2: Price** - This refers to the price of instances in common countries where providers have datacenters;
– **Sub-criterion 1: CPU** - This compares instances in relation to the amount of CPU;
– **Sub-criterion 2: Memory** - This compares instances in relation to the amount of memory;

Table 7 Random consistency index [106]

n	1	2	3	4	5	6	7	8	9	10
RI	0	0	0.58	0.9	1.12	1.24	1.32	1.41	1.45	1.49

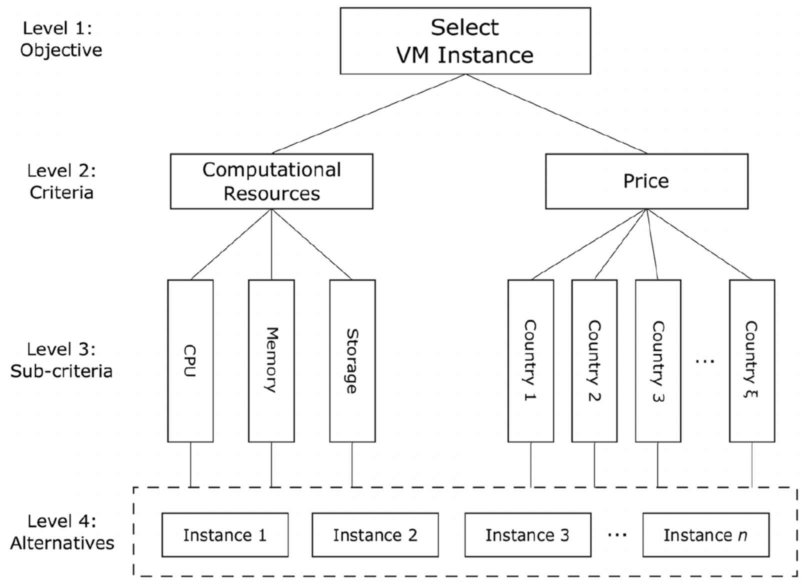

Fig. 3 Hierarchy of D-AHP

– **Sub-criterion 3: Storage** - This compares instances in relation to the amount of storage;
– **Sub-criteria 4** - : Datacenter Location - This compares instances in relation to prices in the ξ countries in which providers have hosted datacenters;
– **Alternatives: Instances** - This refers to the set of selectable instances $I^{\ddagger}$, compared to one another and valued in relation to each decision criterion and subcriterion.

At the end of this step, the selectable instances are classified, from best to worst, according to their respective performance in relation to each criterion and sub-criterion of the hierarchy, considering the weight defined by the DM for each of them in the decision-making process.
In relation to the sub-criteria of the Price criterion, in countries where a given provider has more than one available datacenter, we chose the region within the same country with the minimum value of C_T , which is not necessarily the same for all instances.

5 Application of the methodology

For application of the D-AHP, we defined new application profiles based on actual Cloud usage data contained in the dataset called Google Cluster Trace [113].
This contains, among other information, data relating to computational resources consumed by applications distributed in the form of jobs and tasks. To match the resource metrics of instances, we used GB as a measure of memory and storage resources in the dataset, and CPU resources were measured by cores.
The information in the dataset refers to actual data on the consumption of computational resources by applications running in Google datacenters in the Cloud.
Nonetheless, the data are considered sensitive and for this reason they are obfuscated through a rescaling value before becoming public; the reasons range from economic

aspects to data security [114].
According to Reiss et al. [115], not knowing such factor, by which data were rescaled, makes researchers propose different ways of treating data. In face of this uncertainty, and since a standard is adopted, data can be manipulated in different ways depending on the purpose of use.

5.1 Workload characterization

Zhang et al. [116], in seeking workload models to accurately reproduce the performance characteristics of real workloads, found that simply capturing the average usage of each task would be sufficient to generate synthetic workload with high accuracy, when it comes to the resource usage and task waiting time. Thus, the authors assume that it is possible to realistically estimate the total waiting time and the use of resources for real or imaginary workloads. They came to these conclusions for two reasons: the low variability in the use of resources in the workload by tasks, and the characteristics from evaluation metrics (the use of resources, for instance) under different workload conditions.
In order to generate a realistic and compatible workload with the amounts of computational resources offered by the instances, we propose a rescaling value to be applied to the values related to the total of resources consumed by applications in the dataset, with the aim of turning such values comparable to the resources offered by VM instances. For doing so, we used the second instance with more computational resources among the instances of set I^* as the maximum value for the rescaling so that there are at least two in each decision-making process. In this way, M_{CPU} , M_{MEM} , and M_{STO} refer to the values set for rescaling the CPU, memory, and storage amounts, respectively.
This then gives us a set of jobs $T = \{T_1, T_2,\ldots, T_\Psi\}$, composed of a set of individual tasks $TK =\{tk_1,tk_2,\ldots \ldots,tk_\delta\}$. $R_T =(DC_T , DM_T , DS_T)$ and $R_{tk} =(DC_{tk} , DM_{tk} , DS_{tk})$ are ordered triples whose components are the CPU, memory, and storage resources consumed by a job and by a task, respectively.
The values of the components of R_{tk} can be obtained from the dataset. To obtain the overall values of the components of R_T , we assume that it is calculated by the sum of the values of the resources consumed by the tasks that make up the ψ^{th} job.
According to Reiss et al. [113], applications that need to perform different types of tasks with different resource requirements usually run as multiple jobs. Therefore, let $R_A = (DC_A,DM_A, DS_A)$ denote the amount of computational resources consumed by the workload of the application Ā. The components of R_A can be obtained from the dataset through the sum of resource consumption of the jobs, which, in turn, is obtained through the sum of the consumption of the tasks that compose them.
Thus, an estimate of computational resources $R'_A = (DC'_A , DM'_A , DS'_A)$ consumed by the workload of Ā can be calculated by multiplying the resource consumption $R_A = (DC_A,DM_A, DS_A)$ by its respective rescaling values M_{CPU} , M_{MEM} , and M_{STO} , such that $(DC_A,DM_A, DS_A) \leq 1$.
Therefore, considering the task-job-application relation, we estimate that the computational resource consumption of an application workload from the dataset data is calculated by Eqs. 13–15, as follows:

$$
\begin{aligned}
\mathrm{DC}_A' &= M_{\mathrm{CPU}} \cdot \sum_{T=1}^{\psi} \sum_{tk=1}^{\delta} (\mathrm{DC})_{tk,T} \\
&= M_{\mathrm{CPU}} \cdot [(\mathrm{DC})_{1,1} + (\mathrm{DC})_{2,1} + \ldots + (\mathrm{DC})_{\delta,1} \\
&\quad + (\mathrm{DC})_{1,2} + (\mathrm{DC})_{2,2} + \ldots + (\mathrm{DC})_{\delta,2} + \ldots + (\mathrm{DC})_{\delta,\psi}]
\end{aligned} \tag{13}
$$

$$
\begin{aligned}
\mathrm{DM}_A' &= M_{\mathrm{MEM}} \cdot \sum_{T=1}^{\psi} \sum_{tk=1}^{\delta} (\mathrm{DM})_{tk,T} \\
&= M_{\mathrm{MEM}} \cdot [(\mathrm{DM})_{1,1} + (\mathrm{DM})_{2,1} + \ldots + (\mathrm{DM})_{\delta,1} \\
&\quad + (\mathrm{DM})_{1,2} + (\mathrm{DM})_{2,2} + \ldots + (\mathrm{DM})_{\delta,2} + \ldots + (\mathrm{DM})_{\delta,\psi}]
\end{aligned} \tag{14}
$$

$$
\begin{aligned}
\mathrm{DS}_A' &= M_{\mathrm{STO}} \cdot \sum_{T=1}^{\psi} \sum_{tk=1}^{\delta} (\mathrm{DS})_{tk,T} \\
&= M_{\mathrm{STO}} \cdot [(\mathrm{DS})_{1,1} + (\mathrm{DS})_{2,1} + \ldots + (\mathrm{DS})_{\delta,1} \\
&\quad + (\mathrm{DS})_{1,2} + (\mathrm{DS})_{2,2} + \ldots + (\mathrm{DS})_{\delta,2} + \ldots + (\mathrm{DS})_{\delta,\psi}]
\end{aligned} \tag{15}
$$

5.2 Defining application profiles

Based on dataset values, we propose a set of usage levels based on the interval before rescaling (between 0 and 1). For this, we assume that the usage level of each computational resource (CPU, memory, and storage) by an application Ā can be Low Usage, Medium Usage or High Usage, as shown in Table 8.
Using the values of R_A , we can classify usage levels for the workload of Ā. By combining different usage levels for each of the three computational resources, we can generate a workload profile for Ā, which can be categorized in different ways, as shown in Table 9.

5.3 Case studies

Next, we present three case studies to better understand the proposal to estimate the computational resources consumed by applications from the dataset. From these input data, we apply the D-AHP methodology as a way of proving its effectiveness

Table 8 Categorization of the usage level of computational resources by applications

Usage level (UL)	Value (min–max)
Low usage	$0 \leq \mathrm{UL} \leq 0.3$
Medium usage	$0.3 < \mathrm{UL} < 0.7$
High usage	$0.7 \leq \mathrm{UL} \leq 1$

Table 9 Categorization of the application workload profile

Profile	Description
Very Low	All three resources feature Low Usage
Low	Two resources feature Low Usage, and the third features Medium Usage
Low-to-Medium	One resource features Low Usage, and the rest feature Medium Usage
Low-to-High	One resource features Low Usage, and the rest feature High Usage
Medium	All three resources feature Medium Usage, or all resources feature different usages
High-to-Low	One resource features High Usage, and the rest feature Low Usage
High-to-Medium	One resource features High Usage, and the rest feature Medium Usage
High	Two resources feature High Usage, and the third features Medium Usage
Very High	All three resources feature High Usage

in solving problems that involve selection of VM instances for the migration of applications to the Cloud.

In Case Study 1, we verified the effectiveness of the D-AHP over a reduced set of selectable instances, resulting from an application profile equal to or greater than that classified as Medium. In Case Study 2, we applied the D-AHP to a larger set of selectable instances, obtained through an application profile lower than that classified as Medium. In Case Study 3, we verified the influence of the datacenter location from the results obtained in Case Studies 1 and 2.

5.3.1 Case Study 1

In the first case study, we intended to migrate an application $\bar{A}$ comprising five jobs $T = \{T_1, T_2, T_3, T_4, T_5\}$, each composed of a different number of tasks.

In Table 10, jobs are identified by their JobID. Each of them is composed of a certain number of tasks, according to the dataset. The consumption of resources for each job was obtained through the sum of the consumption of their respective tasks. Table 10 shows that the application has resource usage levels classified as Medium Usage for the CPU, High Usage for Memory, and Low Usage for Storage (see Table 8), according to the values of DC_A , DM_A , and DS_A , respectively. In this way, the workload profile of $\bar{A}$ is classified as Medium (see Table 9).

Using these input data, one can then apply the D-AHP.

Application of Step 1: Step 1 consists of selecting a subset of IaaS providers P^+ belonging to set P by combining the service set offer and Gartner's Magic Quadrant, as described in Subsection 4.1.

Table 10 Consumption of computational resources for the jobs that comprise $\bar{A}$

Job	JobID	Number of tasks	CPU	Memory	Storage
T_1	3418329	3	0.06582	0.27306	0.0010728
T_2	3418368	3	0.12811	0.23804	0.0007467
T_3	28185708	11	0.083154	0.071779	0
T_4	501114088	10	0.0057395	0.004691	0
T_5	5987136072	4	0.33236	0.17517	0.0000391098
Σ			0.6151835	0.76274	0.001858598

Currently, the Leaders' quadrant of Gartner's Magic Quadrant for IaaS providers in 2019 is composed of Amazon (A) [104], Azure (Z) [111], and Google (G)

[117]. In a complementary way, by analyzing the set of established services, it is evident that such providers are the only ones that offer the complete set of services. Thus, the set of selected providers P^+ is composed of three providers, such that $P^+ = \{A, Z, G\}$.

Thus, we can define the set of VM instances of the elements of P^+, such that $IA = \{i_{a,1}, i_{a,2}, \ldots, i_{a,q}\}$, $IZ = \{i_{z,1}, i_{z,2}, \ldots, i_{z,r}\}$, and $IG = \{i_{g,1}, i_{g,2}, \ldots, i_{g,s}\}$ are the sets of all VM instances offered by providers A, Z, and G, respectively.

Among the datacenter location options for providers of set P^+, provider A has datacenters spread across the set of locations $L_A = \{l_{a,1}, l_{a,2}, \ldots, l_{a,\alpha}\}$, while providers Z and G have set of locations $LZ = \{l_{z,1}, l_{z,2}, \ldots, l_{z,\beta}\}$ and $LG = \{l_{g,1}, l_{g,2}, \ldots, l_{g,\gamma}\}$, respectively.

Application of Step 2: In Step 2, Pareto Dominance is applied over the set of instances I_A, I_Z, and I_G for the selected providers. Thus, the intra-provider analysis facilitates comparison of the instances $i_{a,q} \in I_A$, $i_{z,r} \in I_Z$, and $i_{g,s} \in I_G$ in a pairwise way, within their respective sets, in relation to the objectives defined in Eqs. 5–8.

The set of non-dominated instances I^+ resulting from the first phase of dominance application is composed of instances $i^+_{a,q}$, $i^+_{z,r}$, and $i^+_{g,s}$, which refer to providers A, Z, and G, respectively. The number of elements of set I^+ can be changed according to the number of providers selected in Step 1.

After executing the first dominance relationship for each provider's instances, a second dominance relationship is applied to the instances of set I^+. As result, the non-dominated instances selected for the next step of the D-AHP are obtained, which make up the final set of non-dominated instances I^*.

Application of Step 3: In Step 3, the set of datacenter locations is conditioned to the providers selected in Step 1, in order that only instances hosted in all common locations between them are selected.

The elements of sets L_A, L_Z, and L_G are not common to all providers of set P^+; i.e., one provider may have a datacenter in a location where the others do not have a datacenter. However, when we conducted a country-by-country approach, common countries were identified through the intersection of sets L_A, L_Z, and L_G. Thus, when only considering the elements resulting from this operation, the set of selected locations L^+ is composed of the following countries: the USA, Canada, Brazil, England and Ireland, Japan, Australia, Germany, and India.

Regarding definition of the workload application, the values used to rescale each computational resource were as follows: $M_{CPU} = 96$ vCPU, $M_{MEM} = 624$ GB, and $M_{STO} = 6 \times 375 = 2250$ GB. These values correspond to the resources of Google's instance *n1-highmem-96*, which has the second largest amount of resources among all non-dominated instances in set I^*.

To estimate the workload of $\bar{A}$, we multiply the sum of the resource consumption of the jobs obtained in Table 10 by the respective rescaling values. Thus, $DC'_A = 0.6151835 \times 96$, $DM'_A = 0.76274 \times 624$, and $DS'_A = 0.001858598 \times 2250$. Thus, $R'_A = (59.06;475.95;4.18)$.

After calculating R'_A, the set of non-dominated instances I^* is analyzed in order to identify the set of selectable instances $I^\ddagger$, which must have enough resources to support this workload. Based on the R'_A values, the instances in Table 11 were selected, along with their corresponding resources.

Application of Step 4: In Step 4 of the D-AHP, we try to identify the instance that has the best performance in relation to a set of criteria and sub-criteria defined in the hierarchy shown in Fig. 3. For this, weights must be assigned to decision criteria and sub-criteria.

In order to define the weights of the criteria, group decision-making was used. Multiple DMs can contribute a variety of experience, knowledge, and perspectives, and a group can deal with the complexity of the problem better than a single DM.

A questionnaire was presented to a group of professionals (Ω) in the areas of Computing and Software Engineering, whose judgments were grouped using Aggregating Individual Priorities (AIP). In total, 11 DMs obtained consistency in their judgments based on the Consistency Index (CI) and Consistency Ratio (CR) values and, because of this, had their preferences considered. Therefore, $\Omega=\{DM_1, DM_2, \ldots, DM_{11}\}$.

Tables 19 and 20 in Appendix A present the judgments of the 11 DMs in relation to the criteria and sub-criteria of the Computational Resources criterion. These judgments were made according to the Saaty scale.

Note that there was no unanimity in the judgments of all DMs, which may lead, at the end of the process, to selection of different instances. For datacenter locations, equal weights were defined without the influence of DMs to avoid prioritizing instances with a great deal of discrepancy between locations, or those priced more highly than others. Thus, each of the eight locations has a priority equal to 0.125 (or 12.5%).

Table 11 Set of selectable instances with resources equal to or greater than the demand of the application workload

Provider	Instance	vCPU (cores)	Memory (GB)	Storage (GB)
Google	n1-highmem-96	96	624	2250
Amazon	i3.16xlarge	64	488	15.2
	x1.16xlarge	64	976	1920
	x1.32xlarge	128	1952	3840

In relation to decision alternatives, the selectable instances of set $I^{\ddagger}$ are compared to one another according to the actual values of the resources that they have in the form of direct attributes for the sub-criteria linked to the Computational Resources criterion (or benefit criterion); that is, the bigger the better. For sub-criteria linked to the Price criterion, instances are compared with one another again according to the actual prices applicable in each of the countries represented by the sub-criteria, although in the form of indirect attributes (or cost criterion); that is, the smaller the better.

From the data shown in Table 11, the selectable instances are evaluated considering the DMs' preferences in relation to the decision criteria and sub-criteria, as shown in Tables 19 and 20 in Appendix A, in addition to the datacenter location weights (without DMs' preferences).

In the phase prior to applying the weights of the criteria on the selectable instances, the valuation of each one of them is directly related to the amounts of

each resource they have. For example, the instance **x1.32xlarge** is the one that has the most CPU, memory, and storage, and is therefore ranked the best. Instance **i3.16xlarge**, on the other hand, has little storage, which makes its value very low when compared to the others. In relation to the price, because it is an indirect criterion, the instances with the highest price have a lower valuation.
In Table 21 in Appendix A, instance values are shown in relation to the decision criteria, which already account for the values obtained in relation to their respective sub-criteria. In this step, distinct values are noted for different DMs, as per their individual judgments for Computational Resources criterion. With the Price criterion, due to definition of equal weights for all locations, the values obtained were the same for all DMs.
The final classification of the instances, considering the individual judgments of the DMs and after aggregation of their judgments by the AIP, is presented in Table 12. In this case, it is evident that differences in prioritization of criteria and sub-criteria by different DMs result in different classifications of the instances.
For DMs who prioritized the Price criterion over Computational Resources (as in the case of DM_3), the instance with the highest price was the one ranked last, with a value well below the others, considering the maximum importance level (9 on the Saaty scale) attributed by this DM. As a consequence, the lowest-priced instance had the highest score. For the other DMs who did not prioritize any of the criteria, the score for the instances remained close. Such differences may be justified by the different weights assigned to the sub-criteria of the Computational Resources criterion.

Table 12 Final classification of instances for each DM

Instance	Normalized score							Ranking
	$DM_{1,2,6,7,10,11}$	DM_3	DM_4	DM_5	DM_8	DM_9	AIP	
n1-highmem-96	0.262	0.282	0.254	0.253	0.252	0.263	0.257	1
i3.16xlarge	0.229	0.334	0.213	0.247	0.250	0.255	0.253	2
x1.16xlarge	0.229	0.235	0.237	0.226	0.225	0.220	0.239	4
x1.32xlarge	0.280	0.149	0.296	0.274	0.273	0.262	0.251	3

Thus, the Google's instance **n1-highmem-96** obtained the highest global value among the set of selectable instances $I^{\ddagger}$. When analyzing the data in Table 12, it was noted that this instance was only classified as the best by one DM (DM_9), and it was given the second highest classification by all the others. Furthermore, it can be seen that DM_3 judgments significantly influenced the decision of the group because of the larger differences between the evaluations of the instances for this DM in particular. Although some DMs have equal judgments at all levels of the D-AHP hierarchy (in this case, DM_1 , DM_2 , DM_6 , DM_7 , DM_{10} , and DM_{11}) , the aggregation process using the geometric mean method (recommended by Saaty [106]) considers each DM as a member of the group, totaling 11 individual judgments.

5.3.2 Case study 2

In this second case study, we intend to migrate a new application Ā composed of three jobs $T = \{T_1, T_2, T_3\}$, whose resource consumption values are shown in Table 13.

According to Table 13, the application has a resource usage level rated Low Usage for CPU, memory, and storage (see Table 8). In this way, the workload profile of A is classified as Very Low (see Table 9).

Table 13 Consumption of computational resources of the jobs that comprise Ā

Job	JobID	Number of tasks	CPU	Memory	Storage
T_1	5977491124	8	0.0026293	0.006508	0.0000181014
T_2	6238987856	10	0.078815	0.0126	0.000045774
T_3	6239450692	20	0.079183	0.071235	0.00050166
Σ			0.1606273	0.090343	0.0005655354

By applying the rescaling values on DC_A , DM_A , and DS_A , we obtain $DC'_A = 0.1606273 \times 96$, $DM'_A = 0.090343 \times 624$, and $DS'_A = 0.0005655354 \times 2250$. Thus, $R'_A = (15.42;56.37;1.27)$.

Considering that the applications of Steps 1 and 2 are analogous to Case Study 1 (described in Subsection 5.3.1), that is, the providers $\{A, Z,G\} \in P^+$ and set of non-dominated instances I^* are the same, we can directly define the set of selectable instances I‡ on the basis of the R'_A values, as stated in Table 14.

As in Case Study 1, by using the data shown in Table 14, the instances are evaluated considering DMs' preferences (see Tables 19 and 20 in Appendix A). Again, equal weights were defined for all countries for all DMs.

In Table 22 in Appendix A, instances are valued in relation to the decision criteria while already considering the values obtained in relation to their respective subcriteria.

Regarding the sub-criteria of the Computational Resources criterion, once again, the instance **x1.32xlarge** is the one classified as the best since the amount of resources that it possesses is far superior to the majority of other instances. However, due to its higher price, this instance has the worst ranking in relation to the Price criterion.

Table 14 Set of selectable instances with resources equal to or greater than the demand of the application workload

Provider	Instance	vCPU (cores)	Memory (GB)	Storage (GB)
Azure	D16 v3	16	64	400
	E16 v3	16	128	400
	D14 v2	16	112	800
	D15 v2	20	140	1000
	D32 v3	32	128	800
	E32 v3	32	256	800
	D64 v3	64	256	1600
	E64 v3	64	432	1600
Google	n1-standard-16	16	60	1125
	n1-highmem-16	16	104	375
	n1-highmem-32	32	208	375
	n1-highcpu-64	64	57.6	375
	n1-highcpu-96	96	86.4	375
	n1-highmem-64	64	416	375
	n1-standard-96	96	360	375
	n1-highmem-96	96	624	2250
Amazon	i3.16xlarge	64	488	15.2
	x1.16xlarge	64	976	1920
	x1.32xlarge	128	1952	3840

As can be seen from the data in Table 15, most instances have similar classifications due to similar judgments, except for DM_3 , whose judgments prioritize the lowest price.

For DMs who define equal weights of importance for all criteria and sub-criteria (DM_1 , DM_2 , DM_6 , DM_7 , DM_{10} , and DM_{11}), the final classification is characterized by the ratio of the values. This can be verified using the results obtained for the instances **x1.16xlarge** and **x1.32xlarge**. Because **x1.32xlarge** has exactly twice as many computational resources as **x1.16xlarge**, in order for **x1.16xlarge** to be a better option, its price must be less than half the price of **x1.32xlarge**, which was not the case in two of the eight countries analyzed. In addition, just as **x1.32xlarge** has practically double the valuation of **x1.16xlarge** in relation to the Computational Resources criterion, it scores half for the Price criterion.

Regarding the final classification after AIP, Amazon's instance **x1.32xlarge** obtained the highest global value among the set of selectable instances $I^{\ddagger}$. When analyzing the values in Table 15 for each DM, we noted that it was ranked the best by 10 of a total of 11 DMs (i.e., not by DM_3 , who ascribed a higher level of importance to the Price criterion in his judgment. Because of this, instance **D16 v3** was ranked second best.

Table 15 Final classification of instances for each DM

Instance	Normalized score							Ranking
	$DM_{1,2,6,7,10,11}$	DM_3	DM_4	DM_5	DM_8	DM_9	AIP	
D16 v3	0.074	0.120	0.074	0.073	0.073	0.073	0.067	2
E16 v3	0.059	0.091	0.060	0.059	0.059	0.058	0.059	5
D14 v2	0.052	0.074	0.054	0.049	0.049	0.049	0.054	6
D15 v2	0.048	0.060	0.050	0.044	0.043	0.043	0.051	10
D32 v3	0.049	0.062	0.048	0.047	0.046	0.047	0.0521	9
E32 v3	0.044	0.048	0.045	0.043	0.042	0.042	0.04873	11
D64 v3	0.048	0.035	0.047	0.043	0.043	0.045	0.0483	13
E64 v3	0.049	0.030	0.050	0.046	0.046	0.046	0.04870	12
n1-standard-16	0.071	0.105	0.074	0.066	0.065	0.065	0.064	3
n1-highmem-16	0.060	0.094	0.060	0.059	0.059	0.059	0.060	4
n1-highmem-32	0.040	0.050	0.039	0.041	0.041	0.042	0.0476	14
n1-highcpu-64	0.037	0.043	0.031	0.039	0.040	0.045	0.0456	16
n1-highcpu-96	0.036	0.032	0.025	0.040	0.041	0.049	0.044	18
n1-highmem-64	0.037	0.030	0.034	0.043	0.043	0.044	0.045	17
n1-standard-96	0.039	0.027	0.031	0.046	0.047	0.051	0.0458	15
n1-highmem-96	0.060	0.025	0.061	0.057	0.056	0.057	0.0526	7
i3.16xlarge	0.033	0.025	0.030	0.041	0.043	0.042	0.043	19
x1.16xlarge	0.059	0.023	0.066	0.059	0.059	0.052	0.0524	8
x1.32xlarge	0.107	0.026	0.122	0.107	0.107	0.093	0.070	1

5.3.3 Case study 3

In this case study, we intend to analyze the influence of the datacenter location in the selection of VM instances. For this, the same information contained in the Tables used in Case Studies 1 and 2 are used, except for the values referring to the weights of the datacenter locations, which will be modified in order to identify possible classification changes for the instances when defining countries in which providers have datacenters.
To do this, of the eight countries analyzed, different importance levels were defined for three of them: the USA, Brazil, and Japan. The option for the USA is due to the fact that this country has the lowest prices among all the other countries analyzed; Brazil was admitted because it is the same country as the DM group and, consequently, had better latency [8, 21]; Japan was also chosen because of the time zone in relation to Brazil, and applications can always be performed outside peak hours, which usually occur during the day.
The importance levels of these three countries in relation to the rest and to one another, following the Saaty scale, are represented in Table 16.

Table 16 Results of pairwise comparison between sub-criteria of the Price criterion for all DMs

Subcritérios									CR=0.04
	USA	Canada	Brazil	UK	Japan	Australia	Germany	India	Weight
USA	1	5	1/5	5	3	5	5	5	0.201
Canada	1/5	1	1/9	1	1/3	1	1	1	0.042
Brazil	5	9	1	9	7	9	9	9	0.481
UK	1/5	1	1/9	1	1/3	1	1	1	0.042
Japan	1/3	3	1/7	3	1	3	3	3	0.106
Australia	1/5	1	1/9	1	1/3	1	1	1	0.042
Germany	1/5	1	1/9	1	1/3	1	1	1	0.042
India	1/5	1	1/9	1	1/3	1	1	1	0.042

Table 17 Final classification of instances after AIP

Instance	Score	Normalized score	Ranking	Ranking of case study 1	Difference
n1-highmem-96	0.575	0.262	1	1	–
i3.16xlarge	0.546	0.248	3	2	−1
x1.16xlarge	0.526	0.239	4	4	–
x1.32xlarge	0.552	0.251	2	3	+1

From the data shown in Table 23 in Appendix A, in a comparison with Case Study 1 (see Table 12), we verified that the valuation of instances **n1-highmem-96** and **i3.16xlarge** were the ones that underwent the most changes, while the others had less notable changes. However, no instance classifications changed for any of the DMs.

Table 17 shows the final classification of the instances after aggregation of the DMs' judgments. For comparison purposes, two new columns were added to identify the changes in values and, consequently, in the ranking of the instances, when comparing such results with those obtained in Table 12.

By analyzing Table 17, we concluded that the instance **n1-highmem-96** is still classified as the best, although with a greater difference of values over the others. However, it was noted that prioritization of some datacenter locations in relation to others resulted in some changes in the final classification of instances, such as the inversion of classification between instances **i3.16xlarge** and **x1.32xlarge**.

From the data shown in Table 24 in Appendix A, differences were also detected in relation to the results obtained in Table 15, mainly for instances with have an intermediate amount of computational resources, over which the DMs' judgments that did not define equal weights for all elements of the hierarchy in all steps of the pairwise comparisons (i.e., DM_3 , DM_4 , DM_5 , DM_8 , and DM_9) had a greater influence.

Table 18 presents the final classification of the instances after aggregation of the DMs' judgments, together with additional columns, to compare the results with those obtained in Table 15 (see Case Study 2). Based on the values of both tables, we can conclude that, for most instances, despite the final valuation not being the same in both simulations, the classification is maintained.

Table 18 Final classification of instances after AIP

Instance	Score	Normalized score	Ranking	Ranking of case study 2	Difference
D16 v3	0.294	0.067	2	2	–
E16 v3	0.254	0.058	5	5	–
D14 v2	0.245	0.056	6	6	–
D15 v2	0.2281	0.051991	9	10	+1
D32 v3	0.228	0.05193	10	9	−1
E32 v3	0.2095	0.0477	14	11	−3
D64 v3	0.212	0.0483	11	13	+2
E64 v3	0.211	0.0481	12	12	–
n1-standard-16	0.285	0.065	3	3	–
n1-highmem-16	0.265	0.060	4	4	–
n1-highmem-32	0.2099	0.0478	13	14	+1
n1-highcpu-64	0.2014	0.0459	15	16	+1
n1-highcpu-96	0.194	0.044	18	18	–
n1-highmem-64	0.199	0.045	17	17	–
n1-standard-96	0.201	0.0458	16	15	−1
n1-highmem-96	0.231	0.053	7	7	–
i3.16xlarge	0.187	0.043	19	19	–
x1.16xlarge	0.2282	0.051992	8	8	–
x1.32xlarge	0.305	0.070	1	1	–

However, we noticed that, once again, prioritization of some datacenter locations in relation to others caused changes in the final classification of the instances, which can be characterized by a simple inversion of the classification between two instances (as for D15 v2 and D32 v3) or by a more pronounced change (such as E32 v3), which was classified three positions below its classification in Case Study 2.

6 Conclusions

The dynamic pace with which Cloud Computing has been evolving in recent years, providing reliable, affordable, and low-cost computational resources, is driving adoption of the IaaS model. However, there are still many uncertainties surrounding this new paradigm of distributed computing, making a migration process a very complex task. In a market characterized by the presence of multiple providers and the variety of VM instances that each one offers, decisions about the best provider/ instance set make decision-making difficult. In order to help with this problem, we are proposing the D-AHP, a methodology for selecting VM instances in the Cloud, based on Pareto Dominance and the AHP multicriteria optimization method. For this, the D-AHP uses the amount of computational resources and the price of instances in different datacenter locations as decision criteria and subcriteria. A set of new application workload profiles based on the Google Cluster Trace dataset were defined for the case studies presented here, to validate the D-AHP, and these were admitted as migrating to the Cloud. By using the D-AHP, we observed that execution of the Pareto Dominance between instances and filter steps significantly reduces the dimensionality of the problem, as they eliminate instances with less computational resources and have a higher cost if hosted in datacenters from different geographic locations, making the number of pairwise comparisons reduce considerably. The D-AHP method has proved to be efficient because it significantly reduced the number of alternatives to be compared in its last phase, considering that the AHP method is not as efficient when many alternatives are available in the hierarchy, and because with this type of problem it is essential to have the possibility of prioritizing one objective in relation to another in order to meet user needs better. This fact can be verified through different classifications within a set of selectable instances, which are the result of the individual preferences of a set of DMs responsible for the decision process.

In future research, we will endeavor to solve the problem of manual collection of instance details by integrating the D-AHP with existing databases, from which it is possible to obtain information about the prices and computational resource amounts of VM instances belonging to a wide range of providers and, as a result, make the D-AHP an automated tool. In addition, when applying the D-AHP in this study, it was found that, for applications with lower computational demands, the number of selectable instances increases, which can make it difficult to apply the last step of the D-AHP. In order to deal with such situations, we intend, in future research, to seek new alternatives to reduce the dimensionality of the problem, e.g., by adding new criteria to the hierarchy of the D-AHP so that they can be considered in the migration processes of applications with specific demands, such as web applications and integration solutions.

Appendix A: additional tables referring to the case studies

See Tables 19, 20, 21, 22, 23, and 24.

Table 19 Results of pairwise comparison between the criteria for each DM

$DM_{1,2,4,5,6,7,8,9,10,11}$			CI = 0	DM_3			CI = 0
	Resources	Price	Weight		Resources	Price	Weight
Resources	1	1	0.50	Resources	1	1/9	0.10
Price	1	1	0.50	Price	9	1	0.90

Table 20 Results of the pairwise comparison between sub-criteria of the Computational Resources criterion for each DM

$DM_{1,2,6,7,10,11}$				CR=0	DM_3				CR=0.07
	CPU	Mem.	Sto.	Weight		CPU	Mem.	Sto.	Weight
CPU	1	1	1	0.333	CPU	1	1	7	0.515
Mem.	1	1	1	0.333	Mem.	1	1	3	0.388
Sto.	1	1	1	0.333	Sto.	1/7	1/3	1	0.097
DM_4				CR=0	DM_5				CR=0.00
	CPU	Mem.	Sto.	Weight		CPU	Mem.	Sto.	Weight
CPU	1	1/7	1/7	0.066	CPU	1	1	5	0.455
Mem.	7	1	1	0.466	Mem.	1	1	5	0.455
Sto.	7	1	1	0.466	Sto.	1/5	1/5	1	0.091
DM_8				CR=0	DM_9				CR=0.07
	CPU	Mem.	Sto.	Weight		CPU	Mem.	Sto.	Weight
CPU	1	1	9	0.474	CPU	1	3	9	0.655
Mem.	1	1	9	0.474	Mem.	1/3	1	7	0.290
Sto.	1/9	1/9	1	0.053	Sto.	1/9	1/7	1	0.055

Table 21 Valuation of instances in relation to the decision criteria

Instance	$DM_{1,2,6,7,10,11}$		DM_3		DM_4	
	Resources	Price	Resources	Price	Resources	Price
n1-highmem-96	0.236	0.287	0.227	0.287	0.221	0.287
i3.16xlarge	0.101	0.356	0.141	0.356	0.069	0.356
x1.16xlarge	0.221	0.238	0.211	0.238	0.237	0.238
x1.32xlarge	0.442	0.119	0.421	0.119	0.473	0.119
Instance	DM_5		DM_8		DM_9	
	Resources	Price	Resources	Price	Resources	Price
n1-highmem-96	0.220	0.287	0.217	0.287	0.239	0.287
i3.16xlarge	0.138	0.356	0.143	0.356	0.154	0.356
x1.16xlarge	0.214	0.238	0.213	0.238	0.202	0.238
x1.32xlarge	0.428	0.119	0.426	0.119	0.405	0.119

Table 22 Valuation of instances in relation to the decision criteria

Instance	$DM_{1,2,6,7,10,11}$		DM_3		DM_4	
	Resources	Price	Resources	Price	Resources	Price
D16 v3	0.016	0.132	0.014	0.132	0.015	0.132
E16 v3	0.019	0.099	0.018	0.099	0.020	0.099
D14 v2	0.025	0.080	0.019	0.080	0.029	0.080
D15 v2	0.031	0.064	0.023	0.064	0.036	0.064
D32 v3	0.031	0.066	0.028	0.066	0.031	0.066
E32 v3	0.037	0.050	0.035	0.050	0.039	0.050
D64 v3	0.062	0.033	0.056	0.033	0.061	0.033
E64 v3	0.071	0.026	0.066	0.026	0.073	0.026
n1-standard-16	0.028	0.115	0.017	0.115	0.033	0.115
n1-highmem-16	0.017	0.102	0.016	0.102	0.017	0.102
n1-highmem-32	0.027	0.052	0.030	0.052	0.026	0.052
n1-highcpu-64	0.031	0.044	0.038	0.044	0.018	0.044
n1-highcpu-96	0.043	0.029	0.056	0.029	0.022	0.029
n1-highmem-64	0.048	0.026	0.059	0.026	0.042	0.026
n1-standard-96	0.056	0.022	0.072	0.022	0.040	0.022
n1-highmem-96	0.102	0.017	0.097	0.017	0.105	0.017
i3.16xlarge	0.045	0.021	0.061	0.021	0.038	0.021
x1.16xlarge	0.103	0.014	0.098	0.014	0.118	0.014
x1.32xlarge	0.206	0.007	0.197	0.007	0.237	0.007

Instance	DM_5		DM_8		DM_9	
	Resources	Price	Resources	Price	Resources	Price
D16 v3	0.013	0.132	0.013	0.132	0.014	0.132
E16 v3	0.018	0.099	0.018	0.099	0.017	0.099
D14 v2	0.019	0.080	0.018	0.080	0.018	0.080
D15 v2	0.023	0.064	0.022	0.064	0.022	0.064
D32 v3	0.027	0.066	0.026	0.066	0.029	0.066
E32 v3	0.035	0.050	0.035	0.050	0.034	0.050
D64 v3	0.054	0.033	0.053	0.033	0.058	0.033
E64 v3	0.066	0.026	0.065	0.026	0.065	0.026
n1-standard-16	0.018	0.115	0.015	0.115	0.016	0.115
n1-highmem-16	0.016	0.102	0.016	0.102	0.016	0.102
n1-highmem-32	0.030	0.052	0.031	0.052	0.031	0.052
n1-highcpu-64	0.035	0.044	0.035	0.044	0.046	0.044
n1-highcpu-96	0.051	0.029	0.053	0.029	0.068	0.029
n1-highmem-64	0.059	0.026	0.060	0.026	0.061	0.026
n1-standard-96	0.070	0.022	0.072	0.022	0.079	0.022
n1-highmem-96	0.096	0.017	0.095	0.017	0.096	0.017
i3.16xlarge	0.062	0.021	0.064	0.021	0.063	0.021
x1.16xlarge	0.103	0.014	0.103	0.014	0.089	0.014
x1.32xlarge	0.207	0.007	0.207	0.007	0.178	0.007

Table 23 Final classification of instances for each DM

Instance	Normalized score					
	$DM_{1,2,6,7,10,11}$	DM_3	DM_4	DM_5	DM_8	DM_9
n1-highmem-96	0.269	0.295	0.262	0.261	0.260	0.271
i3.16xlarge	0.222	0.322	0.206	0.240	0.243	0.248
x1.16xlarge	0.229	0.234	0.236	0.226	0.225	0.220
x1.32xlarge	0.280	0.149	0.296	0.273	0.272	0.261

Table 24 Final classification of instances for each DM

Instance	Normalized score					
	$DM_{1,2,6,7,10,11}$	DM_3	DM_4	DM_5	DM_8	DM_9
D16 v3	0.073	0.119	0.073	0.072	0.072	0.073
E16 v3	0.056	0.086	0.057	0.056	0.056	0.055
D14 v2	0.055	0.078	0.056	0.051	0.051	0.051
D15 v2	0.049	0.063	0.052	0.045	0.045	0.045
D32 v3	0.048	0.062	0.048	0.046	0.046	0.047
E32 v3	0.042	0.046	0.043	0.041	0.041	0.041
D64 v3	0.047	0.035	0.047	0.043	0.043	0.045
E64 v3	0.048	0.029	0.049	0.045	0.045	0.045
n1-standard-16	0.072	0.107	0.075	0.067	0.066	0.066
n1-highmem-16	0.061	0.095	0.061	0.060	0.060	0.060
n1-highmem-32	0.040	0.051	0.039	0.042	0.042	0.042
n1-highcpu-64	0.038	0.044	0.031	0.040	0.040	0.045
n1-highcpu-96	0.036	0.032	0.026	0.041	0.041	0.049
n1-highmem-64	0.038	0.030	0.034	0.043	0.044	0.044
n1-standard-96	0.039	0.027	0.031	0.046	0.047	0.051
n1-highmem-96	0.060	0.025	0.061	0.057	0.056	0.057
i3.16xlarge	0.033	0.024	0.029	0.0407	0.042	0.041
x1.16xlarge	0.058	0.022	0.066	0.058	0.058	0.051
x1.32xlarge	0.106	0.026	0.122	0.107	0.107	0.092

References

1. Kumar M, Sharma SC (2018) Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment. Comput Elect Eng 69:395–411
2. Smith E, Shirer M (2019) Worldwide public cloud services spending forecast to reach \$160 billion this year, according to idc. https:// www. busin esswi re. com/ news/ home/ 20190 22800 5137/ en/ World wide- Public- Cloud- Servi ces- Spend ing- Forec ast- Reach , 2019. Accessed: 01 May 2020
3. Kumar M, Dubey K, Pandey R (2021) Evolution of emerging computing paradigm cloud to fog: applications, limitations and research challenges. In: 2021 11th International Conference on Cloud Computing, Data Science & Engineering (Confluence), pp 257–261
4. Mell P, Grance T The NIST Definition of Cloud Computing. Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology Gaithersburg, Gaithersburg, special publication 800-145 edition, 2011

5. Hernández I, Sawicki S, Roos-Frantz F, Frantz RZ (2015) Cloud configuration modelling: a literature review from an application integration deployment perspective. Proc Comput Sci 64:977–983
6. Alkhalil A, Sahandi R, John D (2017) A decision process model to support migration to cloud computing. Int J Bus Inf Syst 24:102–106
7. Ramchand K, Chhetri MB, Kowalczyk R (2021) Enterprise adoption of cloud computing with application portfolio profiling and application portfolio assessment. J Cloud Comput Adv Syst Appl 10:1–18
8. Li A, Yang X, Kandula S, Zhang M (2010) CloudCmp: Comparing public cloud providers. In: Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement, pp 1–14
9. Kihal SE, Schlereth C, Skiera B (2012) Price comparison for Infrastructure-as-a-Service. In: Proceedings of the ECIS Conference, pp 1–12
10. Menzel M, Ranjan R (2012) CloudGenius: decision support for web server cloud migration. In: Proceedings of the 21st International Conference on World Wide Web, pp 979–988
11. Mohan Murthy MK, Sanjay HA, Janagal Padmanabha A (2012) Pricing models and pricing schemes of IaaS providers: A comparison study. In: Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp 143–147
12. Menzel M, Ranjan R, Wang L, Khan SU, Chen J (2015) CloudGenius: a hybrid decision support method for automating the migration of web application clusters to public clouds. IEEE Trans Comput 64:1336–1348
13. Emeras J, Varrette S, Bouvry P (2016) Amazon elastic compute cloud (EC2) vs. in-house HPC platform: a cost analysis. In: Proceedings of the CLOUD Conference, pp 284–293
14. López-Pires F, Barán B (2017) Many-objective optimization for virtual machine placement in cloud computing. In: Research Advances in Cloud Computing, Springer, pp 291–326
15. Mitropoulou P, Filiopoulou E, Michalakelis C, Nikolaidou M (2016) Pricing cloud IaaS services based on a hedonic price index. Computing 98:1075–1089
16. Al-Faifi A, Song B, Hassan MM, Alamri A, Gumaei A (2019) A hybrid multi criteria decision method for cloud service selection from smart data. Fut Gen Comput Syst 93:43–57
17. Nagarajan R, Thirunavukarasu R (2019) A fuzzy-based decision-making broker for effective identification and selection of cloud infrastructure services. Soft Comput 23:9669–9683
18. Chauhan N, Agarwal R, Garg K, Choudhury T (2020) Redundant IaaS cloud selection with consideration of multi criteria decision analysis. Proc Comput Sci 167:1325–1333
19. Yao Y, Cao J, Li M (2013) A network-aware virtual machine allocation in cloud datacenter. In: IFIP International Conference on Network and Parallel Computing, pp 71–82
20. Malekimajd M, Movaghar A, Hosseinimotlagh S (2015) Minimizing latency in geo-distributed clouds. J Superc 71:4423–4445
21. Souidi M, Souihi S, Hoceini S, Mellouk A (2015) An adaptive real time mechanism for IaaS cloud provider selection based on QoE aspects. In: 2015 IEEE International Conference on Communications, pp 6809–6814
22. Ziafat H, Babamir SM (2018) Optimal selection of VMs for resource task scheduling in geographically distributed clouds using fuzzy c-mean and MOLP. Soft Pract Exp 48:1820–1846
23. Jamshidi P, Ahmad A, Pahl C (2013) Cloud migration research: a systematic review. IEEE Trans Cloud Comput 1:142–157
24. Kumar M, Sharma SC (2018) PSO-COGENT: cost and energy efficient scheduling in cloud environment with deadline constraint. Sustain Comput Inf Syst 19:147–164
25. Kumar M, Sharma SC, Goel A, Singh SP (2019) A comprehensive survey for scheduling techniques in cloud computing. J Net Comput Appl 143:1–33

26. Leite AF, Alves V, Rodrigues GN, Tadonki C, Eisenbeis C, Melo ACMA (2015) Automating resource selection and configuration in inter-clouds through a software product line method. In: 2015 IEEE 8th International Conference on Cloud Computing, pp 726–733
27. Gómez Sáez S, Andrikopoulos V, Hahn M, Karastoyanova D, Leymann F, Skouradaki M, Vukojevic-Haupt K (2015) Performance and cost evaluation for the migration of a scientific workflow infrastructure to the cloud. In: Proceedings of the CLOSER Conference, pp 352–361
28. Ur Rehman Z, Khadeer Hussain O, Khadeer Hussain F (2013) Multi-criteria IaaS service selection based on QoS history. In: 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA), pp 1129–1135
29. Zhang M, Ranjan R, Nepal S, Menzel M, Haller A (2012) A declarative recommender system for cloud infrastructure services selection. In: International Conference on Grid Economics and Business Models, pp 102–113
30. Nawaz F, Asadabadi MR, Janjua NK, Hussain OK, Chang E, Saberi M (2018) An MCDM method for cloud service selection using a markov chain and the best-worst method. Knowl-Bas Syst 159:120–131
31. Son A-Y, Huh E-N (2017) Study on a migration scheme by fuzzy-logic-based learning and decision approach for QoS in cloud computing. In: 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), pp 507–512
32. Gabsi H, Drira R, Ghezala HHB (2019) A hybrid approach for personalized and optimized IaaS services selection. Int J Adv Int Syst 12:14–26
33. Soltani S, Martin P, Elgazzar K (2018) A hybrid approach to automatic IaaS service selection. J Cloud Comput Adv Syst Appl 7:1–18
34. Zhang G, Zhu X, Bao W, Yan H, Tan D (2018) Local storage based consolidation with resource demand prediction and live migration in clouds. IEEE Access 6:26854–26865
35. Tang J-M, Luo L, Wei K-M (2015) A heuristic resource scheduling algorithm of cloud computing based on polygons correlation calculation. In: 2015 IEEE 12th International Conference on e-Business Engineering, pp 365–370
36. Portella G, Rodrigues GN, Nakano E, Melo ACMA (2018) Statistical analysis of amazon EC2 cloud pricing models. Conc Comp Pract Exp, pp 1–15
37. Erradi A, Sharma B, Bouguettaya A (2017) Using financial options for pricing of IaaS cloud resources. In: 2017 IEEE 10th International Conference on Cloud Computing (CLOUD), pp 584–591
38. Mansouri Y, Nadjaran Toosi A, Buyya R (2013) Brokering algorithms for optimizing the availability and cost of cloud storage services. In: 2013 IEEE 5th International Conference on Cloud Computing Technology and Science, pp 581–589
39. Rodamilans CB, Baruchi A, Midorikawa ET (2014) Experiences applying performance evaluation to select a cloud provider. In: Proceedings of Recent Advances in Computer Engineering, Communications and Information Technology, pp 289–300
40. Chun S-H, Choi B-S (2013) Service models and pricing schemes for cloud computing. Clus Comput 17:529–535
41. Ouarnoughi H, Boukhobza J, Singhoff F, Rubini S (2016) A cost model for virtual machine storage in cloud IaaS context. In: 2016 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP), pp 664–671
42. Gireesha O, Somu N, Krithivasan K, Sriram VSS (2020) IIVIFS-WASPAS: an integrated multicriteria decision-making perspective for cloud service provider selection. Fut Gen Comput Syst 103:91–110

43. Ziafat H, Babamir SM (2017) A method for the optimum selection of datacenters in geographically distributed clouds. J Superc 73:4042–4081
44. Huang J, Kauffman RJ, Ma D (2015) Pricing strategy for cloud computing: a damaged services perspective. Dec Suppl Syst 78:80–92
45. Singh VK, Dutta K (2015) Dynamic price prediction for amazon spot instances. In: 2015 48th Hawaii International Conference on System Sciences, pp 1513–1520
46. Al-Roomi M, Al-Ebrahim S, Buqrais S, Ahmad I (2013) Cloud computing pricing models: a survey. Int J Grid Dist Comput 6:93–106
47. Khajeh-Hosseini A, Greenwood D, Smith JW, Sommerville I (2011) The cloud adoption toolkit: supporting cloud adoption decisions in the enterprise. Soft Pract Exp 42:447–465
48. Samimi P, Patel A (2011) Review of pricing models for grid & cloud computing. In: Proc of IEEE Symposium on Computers and Informatics, pp 634–639
49. Zhao Z, Jiang Y, Zhao X (2015) SLA_oriented service selection in cloud environment: a PROMETHEE_based approach. In: 2015 4th International Conference on Computer Science and Network Technology (ICCSNT), pp 872–875
50. Chen S, Lee H, Moinzadeh K (2019) Pricing schemes in cloud computing: utilization-based vs. reservation-based. Prod Oper Manag 28:82–102
51. Dimitri N (2020) Pricing cloud IaaS computing services. J Cloud Comput Adv Syst Appl 9:1–1152. Jatoth C, Gangadharan GR, Fiore U, Buyya R (2018) SELCLOUD: a hybrid multi-criteria decision-making model for selection of cloud services. Soft Comput 22:1–15
53. Wu C, Buyya R, Ramamohanarao K (2019) Cloud pricing models: taxonomy, survey, and interdisciplinary challenges. ACM Comput Serv 52(108):1–108
54. Baranwal G, Kumar D, Raza Z, Vidyarthi DP (2018) A negotiation based dynamic pricing heuristic in cloud computing. Int J Grid Ut Comput 9:83–96
55. Kansal S, Kumar H, Kaushal S, Sangaiah AK (2018) Genetic algorithm-based cost minimization pricing model for on-demand IaaS cloud service. J Superc 74:1–26
56. Li Y, Meng X, Dong H (2016) A simulated annealing combined genetic algorithm for virtual machine migration in cloud datacenters. In: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), pp 572–577
57. Mazrekaj A, Shabani I, Sejdiu B (2016) Pricing schemes in cloud computing an overview. Int J Adv Comput Sci Appl 7:80–86
58. Ran Y, Yang J, Zhang S, Xi H (2017) Dynamic iaas computing resource provisioning strategy with QoS constraint. IEEE Trans Serv Comput 10:190–202
59. Wang Q, Ming Tan M, Tang X, Cai W (2017) Minimizing cost in IaaS clouds via scheduled instance reservation. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), pp 1565–1574
60. Robert Wilson BM, Khazaei B, Hirsch L (2016) Towards a cloud migration decision support system for small and medium enterprises in Tamil Nadu. In: 2016 IEEE 17th International Symposium on Computational Intelligence and Informatics (CINTI), pp 341–346
61. Zheng W, Xia Y, Zhou M, Wu L, Luo X, Pang S, Zhu Q (2017) Percentile performance estimation of unreliable iaas clouds and their cost-optimal capacity decision. IEEE Access 5:2808–2818
62. Kash IA, Key P, Suksompong W (2019) Simple pricing schemes for the cloud. ACM Trans Econ Comput 7:7:1-7:27
63. Ur Rehman Z, Khadeer Hussain O, Khadeer Hussain F (2012) IaaS cloud selection using

MCDM methods. In: 2012 IEEE Ninth International Conference on e-Business Engineering, pp 246–251
64. Soni A, Hasan M (2017) Pricing schemes in cloud computing: a review. Int J Adv Comput Res 7:60–70
65. Alabool H, Kamil A, Arshad N, Alarabiat D (2018) Cloud service evaluation method-based multi-criteria decision-making: a systematic literature review. J Syst Soft 139:161–188
66. Hosseinzadeh M, Hama HK, Ghafour MY, Masdari M, Ahmed OH, Khezri H (2020) Service selection using multi-criteria decision making: A comprehensive overview. J Net Syst Manag 28:1639–1693
67. Lee S, Seo KK (2013) A multi-criteria decision-making model for an IaaS provider selection problem. Int J Adv Comput Tech 5:363–367
68. Lee S, Seo K-K (2015) A hybrid multi-criteria decision-making model for a cloud service selection problem using BSC, fuzzy delphi method and fuzzy AHP. Wir Per Commun 86:57–75
69. Lee Y-H (2014) A decision framework for cloud service selection for SMEs: AHP analysis. SOP Trans Mark Res 1:51–61
70. Sun M, Zang T, Xu X, Wang R (2013) Consumer-centered cloud services selection using AHP. In: 2013 International Conference on Service Sciences (ICSS), pp 1–6
71. Zhang M, Ranjan R, Menzel M, Nepal S, Strazdins P, Wang L (2015) A cloud infrastructure service recommendation system for optimizing real-time QoS provisioning constraints. IEEE Sys. J., pages 1–12. arXiv preprint arXiv: 1504. 01828
72. Boutkhoum O, Hanine M, Agouti T, Tikniouine A (2016) Selection problem of cloud solution for big data accessing: Fuzzy AHP-PROMETHEE as a proposed methodology. J Dig Inf Manag 14:368–382
73. Meesariganda BR, Ishizaka A (2017) Mapping verbal AHP scale to numerical scale for cloud computing strategy selection. Appl Soft Comput 53:111–118
74. Supriya M, Sangeeta K, Patra GK (2016) Trustworthy cloud service provider selection using multi criteria decision making methods. Eng Lett 24:1–10
75. Sharma M, Sehrawat R (2020) A hybrid multi-criteria decision-making method for cloud adoption: evidence from the healthcare sector. Tech Soc 61:1–12
76. López C, Ishizaka A (2017) GAHPsort: a new group multi-criteria decision method for sorting a large number of the cloud-based ERP solutions. Comput Ind 92:12–24
77. Saripalli P, Pingali G (2011) MADMAC: Multiple attribute decision methodology for adoption of clouds. In: 2011 IEEE 4th International Conference on Cloud Computing, pp 316–323
78. Sohaib O, Naderpour M (2017) Decision making on adoption of cloud computing in e-commerce using fuzzy TOPSIS. In: 2017 IEEE International Conference on Fuzzy Systems (FUZZIEEE), pp 1–6
79. Silas S, Rajsingh EB, Ezra K (2012) Efficient service selection middleware using ELECTRE methodology for cloud environments. Inf Tech J 11:868–875
80. Adamuthe AC, Pandharpatte RM, Thampi GT (2013) Multiobjective virtual machine placement in cloud environment. In: 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies, pp 8–13
81. Malekloo M, Kara N, (2014) Multi-objective ACO virtual machine placement in cloud computing environments. In: 2014 IEEE Globecom Workshops (GC Wkshps), pp 112–116
82. Xu B, Peng Z, Xiao F, Gates AM, Yu J-P (2015) Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. Soft Comput 19:2265–2273

83. Ibrahim E, El-Bahnasawy NA, Omara FA (2016) Task scheduling algorithm in cloud computing environment based on cloud pricing models. In: 2016 World Symposium on Computer Applications & Research (WSCAR), pp 65–71
84. Kumar Sharma N, Reddy Guddeti RM (2016) On demand virtual machine allocation and migration at cloud data center using hybrid of cat swarm optimization and genetic algorithm. In: 2016 Fifth International Conference on Eco-friendly Computing and Communication Systems (ICECCS), pp 27–32
85. Sheikholeslami F, Navimipour NJ (2017) Service allocation in the cloud environments using multi-objective particle swarm optimization algorithm based on crowding distance. Sw Evol Comput 35:53–64
86. Dörterler S, Dörterler M, Ozdemir S (2017) Multi-objective virtual machine placement optimization for cloud computing. In: 2017 International Symposium on Networks, Computers and Communications (ISNCC), pp 1–6
87. Ebadifard F, Morteza Babamir S (2017) Optimizing multi objective based workflow scheduling in cloud computing using black hole algorithm. In: 2017 3th International Conference on Web Research (ICWR), pp 102–108
88. Jahani A, Khanli LM (2016) Cloud service ranking as a multi objective optimization problem. J Superc 72:1897–1926
89. Sofia AS, GaneshKumar P (2018) Multi-objective task scheduling to minimize energy consumption and makespan of cloud computing using NSGA-II. J Net Syst Man 26:463–485
90. Tao F, Li C, Liao TW, Laili Y (2016) BGM-BLA: a new algorithm for dynamic migration of virtual machines in cloud computing. IEEE Trans Serv Comput 9:910–925
91. Guo L, He Z, Zhao S, Zhang N, Wang J, Jiang C (2012) Multi-objective optimization for data placement strategy in cloud computing. In: International Conference on Information Computing and Applications, pp 119–126
92. Ramezani F, Lu J, Taheri J, Zomaya AY (2017) A multi-objective load balancing system for cloud environments. Comput J 60:1316–1337
93. Zuo L, Shu L, Dong S, Zhu C, Hara T (2015) A multi-objective optimization scheduling method based on the ant colony algorithm in cloud computing. IEEE Access 3:2687–2699
94. Fang F, Qu B-B (2017) Multi-objective virtual machine placement for load balancing. In: Proceedings of the IST Conference, pp 1–9
95. Chen J, Qin Y, Ye Y, Tang Z (2015) A live migration algorithm for virtual machine in a cloud computing environment. In: 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), pp 1319–1326
96. Baranwal G, Prakash Vidyarthi D (2014) A framework for selection of best cloud service provider using ranked voting method. In: 2014 IEEE International Advance Computing Conference (IACC), pp 831–837
97. Chung BD, Seo K-K (2015) A cloud service selection model based on analytic network process. Ind J Sci Technol 8:1–5
98. Chang X, Wang B, Muppala JK, Liu J (2016) Modeling active virtual machines on IaaS clouds using an M/G/m/m+K queue. IEEE Trans Serv Comput 9:408–420
99. Kumari A, Jain S (2016) Auction based resource allocation strategy for infrastructure as a service. In: 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), pp 381–386

100. de Moraes LB, Fiorese A, Matos F (2017) A multi-criteria scoring method based on performance indicators for cloud computing provider selection. In: Proceedings of the ICEIS Conference, pp 588–599
101. Rui Z, Bingyong T (2016) The pricing of cloud computing with preferential policies. In: 2016 IEEE 13th International Conference on e-Business Engineering (ICEBE), pp 232–237
102. Vukovic M, Hwang J (2016) Cloud migration using automated planning. In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium, pp 96–103
103. Marks EA, Lozano B (2010) Executive's guide to cloud computing. John Wiley & Sons Inc, Hoboken
104. Amazon. Amazon web services. https:// aws. amazon. com/, 2018. Accessed: 01 November 2018
105. Pareto V (1896) Cours D'Économie Politique. F. Rouge
106. Saaty TL (1980) The analytic hierarchy process. McGraw-Hill, New York
107. Kumar M, Kishor A, Abawajy J, Agarwal P, Singh A, Zomaya A (2021) ARPS: An autonomic resource provisioning and scheduling framework for cloud platforms. IEEE Transactions on Sustainable Computing
108. Saha M, Panda SK, Panigrahi S (2021) A hybrid multi-criteria decision making algorithm for cloud service selection. Int J Inf Tech 13:1417–1422
109. Bala R, Gill B, Smith D, Wright D (2020) Magic quadrant for cloud Infrastructure as a Service, worldwide. https:// www. gartn er. com/ doc/ repri nts? id=1- 1CMAP XNO& ct= 19070 9& st=sb, 2019. Accessed: 15 January 2020
110. Cloudorado (2017) Cloud Computing Comparison Engine. https:// www. cloud orado. com/. Accessed: 15 December 2017
111. Azure (2018) Microsoft azure. https:// azure. com/, Accessed: 10 November 2018
112. Saaty TL (1990) How to make a decision: the analytic hierarchy process. Eur J Oper Res 48:9–26
113. Reiss C, Wilkes J, Hellerstein JL (2011) Google cluster-usage traces: format + schema. Google Inc., Technical report
114. Reiss C, Wilkes J, Hellerstein JL (2012) Obfuscatory obscanturism: Making workload traces of commercially-sensitive systems safe to release. In: 2012 IEEE Network Operations and Management Symposium, pp 1279–1286
115. Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA (2012) Towards understanding heterogeneous clouds at scale: google trace analysis. Intel Science & Technology Center for Cloud Computing, Technical report
116. Zhang Q, Hellerstein JL, Boutaba R (2011) Characterizing task usage shapes in google's compute clusters. In: Proceedings of the LSDSM Workshop, pp 1–6
117. Google (2018) Google compute engine. https:// cloud. google. com/. Accessed: 12 November 2018