The Journal of Supercomputing (2019) 75:6324–6360
https://doi.org/10.1007/s11227-019-02854-1

A survey and taxonomy on virtual data center embedding

Ameni Hbaieb[1] · Mahdi Khemakhem[2,3] · Maher Ben Jemaa[1]

Published online: 15 April 2019

Abstract

Data center network virtualization is being considered as a promising technology to provide a performance guarantee for cloud computing applications. One important problem in data center network virtualization technology is virtual data center (VDC) embedding, which handles the physical resource allocation to virtual nodes (virtual switches and virtual servers) and virtual links of a VDC. When node and link constraints (including CPU, memory, storage and network bandwidth) are both taken into account, the VDC embedding (VDCE) problem becomes NP-hard. The VDCE is so crucial that took wide consideration since it directly affects the execution, resource use and power consumption of data centers. To the best of our knowledge, there is no published work that precisely outlines open challenges connected with VDCE problem including all of its variants. On this point, this work tries to articulate this problem and bring research taxonomy for succinct classification of existing works. Moreover, we summarize the possible techniques already presented in the literature and we establish a classification based on a taxonomy study. At the end, we examine the limitations of existing solutions and identify the related open challenges.

Keywords Data center network virtualization · Cloud computing · Virtual data center embedding · Embedding algorithms · Resource allocation

✉ Mahdi Khemakhem
m.khemakhem@psau.edu.sa; mahdi.khemakhem@enetcom.usf.tn

Ameni Hbaieb
ameni.hbaieb@redcad.tn

Maher Ben Jemaa
maher.benjemaa@redcad.tn

[1] ReDCAD Lab. - LR13ES26, National School of Engineers, University of Sfax, Sfax, Tunisia

[2] College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University, Al-Kharj, Saudi Arabia

[3] LOGIQ Lab. - UR13ES71, National School of Electronics and Telecoms of Sfax, University of Sfax, Sfax, Tunisia

1 Introduction

The Internet has made significant impact on our society, business models and our daily lives. As a new paradigm for the future Internet, cloud computing has drawn the attention of the general public in recent years. In simple terms, cloud computing is able to create a virtual environment which allows both software and hardware to be shared by multiple users via Internet. With cloud computing, businesses are no longer required to incur investment on purchasing hardware and software licenses in order to deploy their services and applications. Moreover, human expenses can also be reduced since the operating and maintaining cost will be shifted to the cloud side [19].

Among different cloud delivery models, the infrastructure as a service (IaaS) is typically delivered to users as virtual machine (VM) instances with different resource configurations and quality of service (QoS) guarantees. The emergence of server virtualization technologies has allowed multiple VMs to coexist on the same physical host and operate them independently of each other. An infrastructure provider (InP) depending on server virtualization is able to create, manage and allocate resources for virtual components. By sharing infrastructure, the cost of operating and equipment investment can be greatly reduced.

To ensure meeting the QoS to users, it is necessary that VMs are efficiently mapped to given physical machines (PMs). The process of mapping VMs to PMs is known as virtual machine placement (VMP). Obviously, VMP is one of the major issues in cloud computing [50].

However, only *server virtualization* is not enough to address the entire limitations of today data center architectures. Particularly, data center networks still suffer from a number of limitations pertaining to the manageability, network performance and security. Motivated by these restraints, there is an emerging trend toward fully virtualizing data center networks in addition to server virtualization [6]. So far, most of the existing work on network virtualization has been focused on virtualizing traditional Internet service provider (ISP) networks. Thus, virtualizing data center networks is a relatively new research direction and a key step toward fully virtualized data center architectures.

In a virtualized data center (VDC), some or all of the hardware (e.g., servers, routers, switches and links) are virtualized. As a result, recent study suggestions have supported the idea of offering both computing and networking resources in the form of virtual data centers (VDCs). Basically, a VDC is a logical instance of a virtualized data center consisting of a subset of the physical data center resources. One of the main challenges associated with the management of cloud computing data center resources is the problem of allocating resources to VDCs, known as the *virtual data center embedding (VDCE) or mapping problem*, which aims to find an efficient mapping of virtual components to the physical components (e.g., servers, routers, switches and links) that have limited resources. Since various VDCs share the physical resources, mapping of VDCs becomes critical as efficient and effective integration solutions can cost less and result in a better QoS. Therefore, it is substantial for infrastructure providers to optimize the embedding of users' requests to improve their income.

In the literature, the survey of [6] presents data center virtualization frameworks, particularly the enabling data center architectures. In addition, the authors compare different proposals using a set of qualitative metrics including: scalability, fault tolerance, deployability, QoS support and load balancing. Some surveys have been conducted on the VDCE such as the work presented in [13, 61]. In [61], the authors focused on the embedding techniques and compared their functionalities. In [13], written in Spanish language, a taxonomic classification was suggested. Besides, the authors discussed the future research directions in terms of type of algorithms, fault tolerance, distributed allocation and ecological allocation. On the other hand, our work goes beyond what the aforementioned papers provide: (1) The VDCE problem in all its variants is discussed and current approaches proposed by academic community are categorized. (2) A more comprehensive survey of the literature is presented. (3) We discuss well-known problems that are closely related to the VDCE problem, from which are worth drawing inspiration.

The rest of this paper is organized as follows: Section 2 articulates the key concepts concerning the issue of VDCE problem. Section 3 presents the taxonomy based on research dimensions for classifying the existing literature in the concerned research area. Section 4 classifies the existing state of the art based on proposed research taxonomy. Section 5 presents the summary and our plan for future work. The paper ends with some conclusive remarks in Sect. 6.

2 Key concepts

This section presents the concepts connected to embedding cloud computing virtual data center that forms the fundamentals of our research work.

2.1 Business model

Unlike the traditional networking model, data center virtualization separates the role of the traditional ISP into two: an infrastructure provider (InP) and a service provider (SP) [6]. Hence, it separates the role of deploying networking mechanisms, i.e., protocols, services (i.e., SP) from the role of owning and maintaining the physical infrastructure (i.e., InP). Besides being the company that owns and manages the physical infrastructure of a data center, an InP leases virtualized resources to multiple service providers/tenants. Each tenant creates a VDC over the physical infrastructure owned by the InP for further deployment of services and applications offered to end users. Thus, several SPs can deploy their coexisting heterogeneous network architectures required for delivering services and applications over the same physical data center infrastructure.

2.2 Virtual data center

Usually, VMs are used by clients to accomplish certain jobs, frequent portions of a larger application. For instance, we consider a three-tier web application that is composed of: a web server, an application server and a database server. Data transfer between the different servers is a part of the architecture. If the exchange of data between the application server and the database server fails, it may result in an undesirable growth in response time or latency along with raised energy consumption in the concerned hardware elements. Additionally, all the services turn inaccessible despite the communication between web servers and databases. In this manner, it is essential to support the dependencies between entities in the virtual model of virtual data centers in terms of bandwidth [52].

Motivated by this observation, currently, a considerable number of research papers suggest a better approach of allocating resources in the form of virtual data centers instead of VMs [28]. Essentially, a VDC represents a client request to deploy an application in a cloud data center. It is composed of virtual machines, routers and switches connected through virtual links. Each virtual link is described by its bandwidth capacity, propagation delay and so on. Hence, this allows infrastructure providers to achieve better performance isolation between VDCs and to implement more fine-grained resource allocation schemes. Meanwhile, VDCs allow service providers to guarantee predictable network performance for their applications.

Figure 1 shows two examples of VDC requests. The first example of virtual data center request represents an application based on the well-known three-tier architecture. Three-tier architecture is a client–server architecture in which different components are developed and maintained as independent modules on separate platforms. In this example, the user interface, functional process logic and data access are connected throw a guaranteed bandwidth. Each tier could involve numerous servers for availability, replication or balancing reasons. The second VDC request considers an example of a parallel processing application. The data are partitioned across multiple parallel computing slave nodes with each node having its own hardware to process the data locally. Meanwhile, the master node maintains and manages the nodes in the cluster. In this architecture, all communications happen via local network.

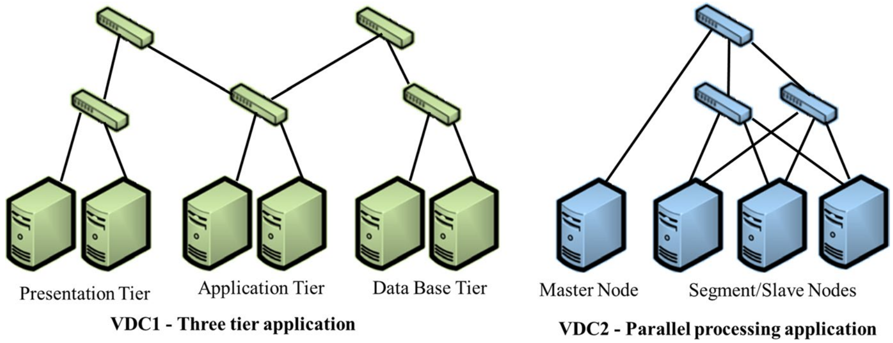

Fig. 1 Examples of VDC requests

2.3 What is VDCE problem?

The goal of resource allocation for any specific cloud service provider is to maximize profit by either optimizing applications QoS, in order to avoid violation of the service-level agreements (SLAs) [8]. A violation is interpreted into a punishment that he has to pay and maximizing resource utilization, or improving energy efficiency. However, the goal of cloud user is to minimize payment by renting the resources.

VDCE problem is all about integrating cloud provider activities for utilizing and allocating resources taking into account the restrictions of cloud data center capacity in order to fulfill the demands of the customers. It needs the type and amount of resources required by each application so as to complete a user job, where resources are often outlined as CPU, memory, disk, network bandwidth and delay. The order and time duration of allocation of requests are also an input for an optimal assignment.

A cloud owner's resource management actions toward simultaneously maximizing revenue, minimizing resource wastage and enhancing QoS can be classified as follows [35]:

- *Initial embedding* It deals with continuous arrival of VDC deployment requests at runtime. Quick response is a crucial metric for ensuring a high service quality of the embedding.
- *Overload migration* When physical resources are not enough to answer the running virtual resources demands, virtual nodes/links can be migrated to a different host to make available the needed resources.
- *Consolidation* In this context, various virtual machines are hosted on the minimal possible physical servers, and remainder of the needless resources are switched off to sleep mode. The reduction in the number of servers has a noticeable benefit on data centers.
- *Load balancing* The goal is to balance the distribution of load across data center, thus avoiding a situation where there is a big discrepancy in resource usage levels and increasing the number of physical resources available for hosting other VDCs.

2.4 Related embedding problems

At first glance, the VDCE problem is not different from the traditional virtual network embedding (VNE), virtual cluster embedding (VCE) and service function chain (SFC) embedding. In fact, VDCE problem is tightly connected to these problems, since they have the similarity that they try to place virtual objects on appropriate location, which can be considered as a variant of the resource allocation problem. However, each problem focuses on different applications and use cases. In this section, we will summarize the similarities and differences of the previously mentioned embedding problem variants.

2.4.1 Virtual network embedding

Similar to server virtualization, network virtualization aims at creating multiple virtual networks (VNs) on top of a shared physical substrate network (SN) allowing each VN to be implemented and managed independently [12, 18]. A VN is a combination of active and passive network elements (network nodes and network links) where virtual nodes are interconnected through virtual links, forming a virtual topology. By virtualizing both node and link resources of a SN, multiple virtual network topologies with widely varying characteristics can be created and coexist on the same physical hardware. This leads to the problem of how the virtual networks be realized by the physical networks, which is the main challenge in network virtualization and is called virtual network embedding (VNE) problem. As we mentioned, both VN and SN abstractions are graphs that consist of nodes and links. Therefore, to some extent, VNE problem is finding a subgraph in the physical network topology that is isomorphic with the virtual network topology.

Relationship between VDCE and VNE Both VNE and VDCE problems rely on virtualization techniques to partition available resources and share them among different users. However, multiple features make the VNE and VDCE different problems including: (1) While virtualized ISP (VNs) networks generally consist of packet forwarding elements (e.g., routers), virtualized data centers include diverse types of nodes including servers, routers, switches and storage nodes. Hence, unlike a VN, a VDC is composed of various types of virtual nodes (e.g., VMs, virtual switches and virtual routers) with different resources (e.g., CPU, memory and disk). (2) In addition, in the context of network virtualization, virtual links are characterized by their bandwidth. Propagation delay is an essential metric when nodes are geographically distributed. Nevertheless, since a data center network covers a small geographic area, the propagation delay between nodes is negligible; hence, it is always ignored when defining VDC virtual links [5, 28]. (3) Furthermore, different from ISP networks, data center networks are built using topologies like the conventional tree, fat tree, or Clos topologies with well-defined properties, allowing to develop embedding algorithms optimized for such particular topologies. In summary, data center network virtualization is different from ISP network virtualization, because one has to consider different constraints, resources and specific topologies.

2.4.2 Service function chain placement

An operator network consists of a large number of intermediate network functions (NFs). Network address translators (NATs), load balancers, firewalls and intrusion detection systems (IDSs) are examples of such functions. Traditionally, these functions are implemented on physical middle boxes, based on special purpose hardware platforms that are expensive and difficult to maintain and upgrade. Following the trend of virtualization in large-scale networks, network functions that were deployed as middle boxes are also being replaced by *virtual network functions (VNFs)*. Typically, network flows go through several network functions. That means a set of NFs is specified and the flows traverse these NFs in a specific order so that the required functions are applied to the flows. This notion is known as service function chaining

(SFC) [55, 60]. The challenge is hence to find the best placement for the function chains in network function virtualization infrastructure, considering the requirements of individual requests as well as the overall requirements of all network applications and the combination of requests. The problem is known as *service function chain placement*.

Relationship between VDCE and SFC placement Even if SFC placement seems to be similar to the VDCE problem, they are quite different due to the following features: (1) In SFC placement problem, traffics must flow through predefined ordered network functions. However, VDCE problem do not have such order requirement. (2) Moreover, forwarding latency behavior is a specific aspect of SFC placement that is not addressed in VDCE problem. Depending on service characteristics, SFCs have different latency requirements. For example, the chain for the latency-sensitive service such as the multimedia streaming and voice over IP (VoIP) has a tight latency requirement, while the chain for the file download service has a loose latency requirement. (3) Additionally, virtual network functions can be shared by multiple flows. However, different VDCs are typically independent, i.e., a flow of a VDC A do not traverse through the virtual machines/switches of another VDC B. (The flow may traverse the physical resource that host the virtual machine/switch of B.)

2.4.3 Virtual cluster embedding

Early works on virtualized data center have modeled the virtual network connecting the virtual machines as a *virtual cluster* (VC) [5, 68]. A virtual cluster is a star-shaped virtual network topology connecting virtual machines to a logical switch with absolute bandwidth guarantees. This virtual network abstraction is attractive for its simplicity and its flexibility since it supports all communication patterns. The question of how to embed a virtual cluster in a given data center such that it consumes a minimal amount of resources while providing its performance guarantees is an interesting algorithmic problem called: *virtual cluster embedding* (VCE).

Relationship between VDCE and VCE The VC embedding problem is similar to the VDCE problem. However, the major difference between them is the virtual infrastructure. While the VDC is composed of different types of virtual nodes (e.g., VMs, virtual switches and virtual routers), the VC is restricted to a set of VMs and a virtual switch connecting them; thus, a VC is a part of a VDC. Hence, the VCE is a special case of the generalized VDCE problem which will be possible to solve it by using an algorithm designed for VDCE problem.

2.5 Problem formulation

Regularly, the VDCE problem can be defined according to [56] as follows:

The physical data center is expressed by a graph $G^p = (M^p, S^p, L^p)$ where M^p is the set of physical servers, S^p is the set of physical switches and L^p is the set of physical links. Both physical nodes and links have requirements. Node requirements can be CPU, RAM, geographic location, etc. However, link requirements can be bandwidth, delay, etc.

The virtual data center request consists of virtual machines, virtual switches and virtual links, also expressed by a graph $G^v = (M^v, S^v, L^v))$ with requirements that represents the demands of the virtual machines and links. The mapping of virtual nodes and links onto the physical infrastructure is performed by an embedding algorithm.

The VDCE problem can usually be divided into at least three types of separate mappings: virtual server mapping, virtual switch mapping and virtual link mapping.

- Virtual machines mapping: $\Gamma_M : M^v \mapsto M^p$
  In this first stage, the goal is to assign each VM in M^v into exactly one physical machine in M^p. The mapping must obey the capacity of hardware; that is, the physical machine that accommodates a VM must have enough capacity.
- Virtual switches mapping: $\Gamma_S : S^v \mapsto S^p$
  The second stage is about mapping each virtual switch in S^v to a physical switch in S^p. Along with the hardware capacity constraint, the major idea here is to assign virtual switches to physical ones while considering the former VM placement.
- Virtual link mapping: $\Gamma_L : L^v \mapsto L^p$
  Finally, each virtual link is mapped either to a physical path (unsplittable flow) or a set of physical paths (splittable flow) in the middle of the corresponding physical machines/switches that accommodate the end virtual machines/switches of that virtual link.

Figure 2 depicts a scenario of an embedding process. The physical and virtual nodes are labeled with letters inside the corresponding machine/switch. We have the embedding of G^v on G^p as Γ, where Γ is defined as follows: $\Gamma_M(a) = (A)$, $\Gamma_M(b) = (B)$, $\Gamma_M(c) = (C)$, $\Gamma_S(d) = (E)$ and $\Gamma_L(ad) = \{AE\}$, $\Gamma_L(bd) = \{BE\}$, $\Gamma_L(cd) = \{CF, FG, GE\}$.

VDCE problem complexity Solving the VDCE problem is NP-hard, as it considers simultaneously two NP-hard problems: embedding of virtual links and virtual machines. In fact, a sufficiently general formulation of the VM placement problem includes the well-known bin-packing problem [47] as a special case, and hence, it is strongly NP-hard. Even the allocation of a virtual link set to single physical path such as the virtual link embedding is also NP-hard since it is considered as a reduction from the unsplittable flow problem [14].

3 Taxonomy of VDCE approaches

As we mentioned earlier, there is a number of virtual data center approaches that has not yet been surveyed. We were guided by the literature to consider the dimensions of our taxonomy. By reading the state-of-the-art papers, we were able to explicitly extract the highlighted dimensions. To do so, this section makes a comparison based on the following dimensions: data center network topology,

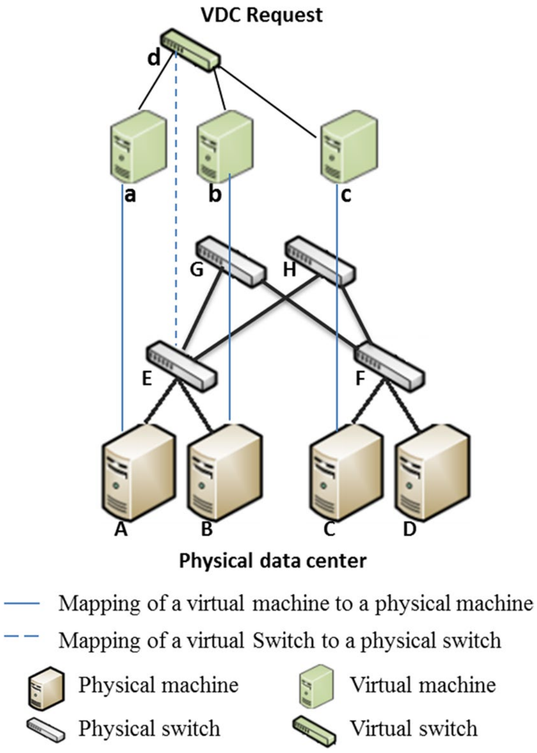

Fig. 2 Mapping of VDC request onto a shared data center

cloud architecture, problem decomposition, embedding approaches, objective function and solution techniques.

3.1 Data center network topology

Different network architectures for data centers have been considered and achieved in both academia and industry. This section targets single-root tree topology, Clos topology and fat tree topology, which are frequently adopted to evaluate cloud resource allocation mechanisms.

3.1.1 Conventional data center network topology

Modern data centers pursue to a great extent a frequent network architecture, known as the three-tier architecture [34]. At the basic level, known as the access tier, each server attaches to one (or two, for redundancy function) access switch. Each access switch connects to one (or two) switches at the aggregation tier, and finally, each aggregation switch attaches with numerous switches at the core tier.

3.1.2 VL2

VL2 [25] is a new architecture that shares many aspects with the earlier one. More precisely, it is a three-tier architecture with major distinction that the core tier and the aggregation tier that form a Clos topology [15], i.e., the aggregation switches are linked with the core ones by creating a complete bipartite graph.

3.1.3 Fat tree

Fat tree [43] is a special category of Clos topology that is defined in a treelike construction. The topology is based on the concept of pods: It is constituted of k-port switches involving k pods. Each pod contains two layers (aggregation and edge) of $k/2$ switches. Each of $(k/2)^2$ core switches has one port attached to each of the k pods.

3.1.4 BCube

BCube [27] is a new multi-level network architecture for the data center with the following distinguishing feature: Servers are part of the network infrastructure, i.e., they forward packets on behalf of other servers. BCube is a recursively defined structure. At level 0, $BCube0$ consists of n servers that connect together with an n-port switch. A $BCube_k$ consists of n $BCube_{k-1}$ connected with n^k n-port switches.

3.1.5 Star

It is one where there is a central switch connected to other devices. The switch must have n ports to support n servers. Generally, they are used to represent virtual requests and not to build a physical network supporting a data center.

3.2 Cloud architecture

The VDCE problem has been studied in two different cloud architectures: single cloud and multi-cloud.

3.2.1 Single cloud

The VDCE problem within a single data center has been largely studied in the literature. It considers the case where all the VDC units are allocated within the same data center. In this case, the InP has a single DC with a number of physical resources. Generally, the number of servers is expected to be high enough to serve all customers requests. Classical objectives are minimizing the utilization of resources and optimizing global energy consumption, subject to performance restraints (SLAs). For more details, the reader may refer to [28, 56, 57, 70, 72, 74, 76, 77].

3.2.2 Multi-cloud

There are only negligible number of researches that have examined the problem of VDCE within geographically distributed DCs. Multi-cloud embedding of VDCs considers a distributed infrastructure consisting of multiple data centers located in different regions and interconnected through a backbone network. In such a process, as shown in Fig. 3, VDC components are divided into several parts and assigned to different physical data centers. Papers taking into account such architecture are due to [2, 3, 30].

In such an environment, since users may require services from various locations, traffic between DCs is very significant. Obviously, the essential problem in the process of decision making is calculating the optimal mapping schema for several VDCs while ensuring application performance and QoS. However, providers are also challenged by minimizing the inter-DCs traffic made by the various

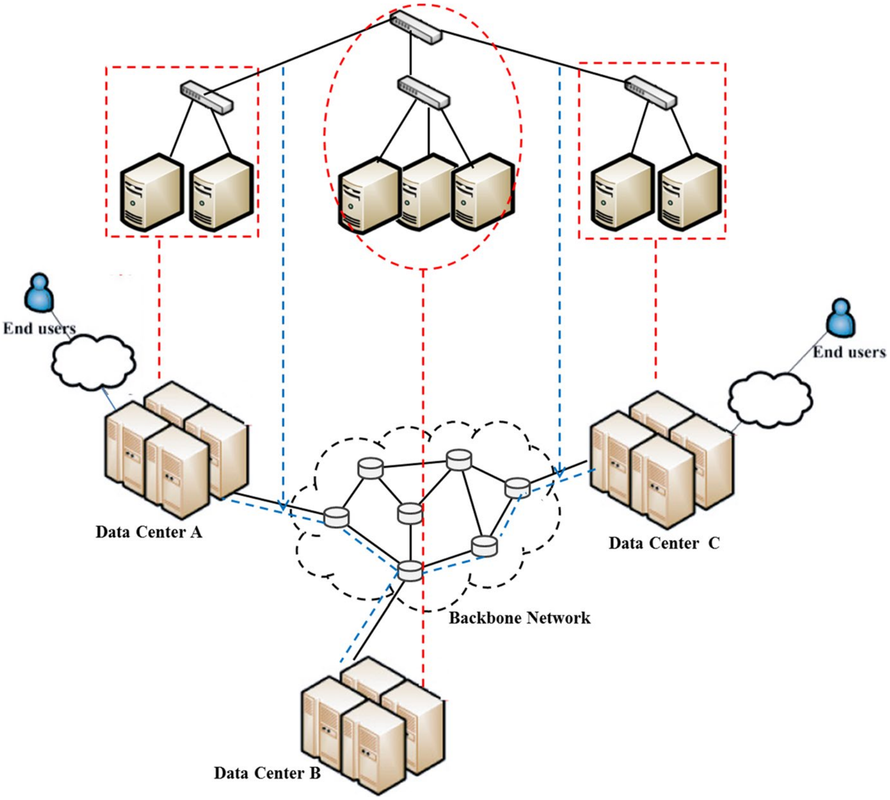

Fig. 3 Example of VDC deployment over a multi-cloud architecture (source: [3])

communicating VMs belonging to the same VDC to avoid possible network congestion problems and reduce energy consumption of DCs. This interest is due to the attempt to obtain a trade-off between operational costs and performance.

3.3 Problem decomposition

In this work, we clearly distinguish three types of decompositions of the VDCE problem: one stage, multistage and inter-InP. We elaborate on these decompositions in the next subsections.

3.3.1 One-stage decomposition

Some approaches have used one-stage decomposition to solve VDCE. They attempt to solve the three subproblems simultaneously. This decomposition was achieved by [28, 72, 74].

3.3.2 Multistage decomposition

As already been mentioned in the formulation Sect. 2.5, the VDCE problem is solved when its three embedding phases: VM mapping, switch mapping and link mapping are solved. The multistage embedding starts by VM mapping stage as it provides the input to solve the next stage. Lately, some approaches have proposed to solve virtual switches and links in different stages, while others intend to jointly embed virtual switches and links in one stage. Some examples for this variant are the works proposed in [56, 57, 70, 77].

3.3.3 Inter-InP decomposition

Taking into account different InPs, we can decompose the embedding across several InPs into a set of single-InP problems. Obviously, at the arrival of a VDC request, the first stage aims at splitting it into several partitions. Hence, each partition is then embedded in different data centers and connected through an external links among InPs. Inside each InP, the sub-requests belonging to the same VDC are mapped using ordinary VDCE algorithms; see [2, 3].

3.4 Embedding approaches

As it is proposed by [18] to classify VNE approaches, all VDCE approaches presented in the literature could be classified according to whether *static* or *dynamic* and *concise* or *redundant*.

3.4.1 Static versus dynamic

VDCE approaches can be classified as dynamic and static; however, in most real-world situations, it must be approached as an online problem which stems from two major factors. First of all, the amount of physical resource is variable throughout time due to the possibility of addition, removal and temporary unavailability of hardware/software in a data center. Additionally, users requests come and stay in the system for an arbitrary amount of time without knowing in advance the changement of the resources requirements [58]. Over time, the VDC requests leaving the system are continuously releasing resources.

- Dynamic approaches attempt to reconfigure VDC requests already mapped in order to reorganize the distribution of resources and optimize the use of infrastructure resources. The main idea in these approaches is to use live migration [53], whereby running virtual resources are paused, serialized and transferred to a different host, where they are once again scheduled for execution. Live migration offers many benefits, including maximizing fault tolerance, simplifying system maintenance and, from the resource management view, ensuring global embedding objectives, such as load balancing and consolidating virtual components on a minimal number of hosts. For more details, readers can refer to [21, 28, 30, 72, 74].
- Static approaches do not provide the possibility of remapping one or more VDC. In fact, the unpredictable and changing nature of users' requests in these approaches is ignored and the mapping of the VDCs is fixed throughout the lifetime of the request and it will not be recalculated for a long period of time. For more details, readers can refer to [3, 56, 70, 77].

3.4.2 Concise versus redundant

Survivability has lately been an important issue for network virtualization. Since virtualization enables running multiple virtual resources on a shared physical infrastructure, a failure in the physical component can disturb other virtual resources that are mapped to it; therefore, the survivability needs to be considered in the process of embedding virtual resources to ensure required VDC availability. To do that, VDCE problem can be formulated as concise or redundant problems.

- A redundant approach is mostly a protection method for physical device breakdown, which reserves fall-backup resources before any failure happens. The main challenge in this approach is to compute the number and determine the placement of redundant virtual resource provisioned as backups, with the goal of maximizing VDC availability; see [57, 72]. Embedding solutions that consider multi-path virtual link embedding are an example for this strategy [76].
- Concise approaches, however, only use as much physical necessary resources to satisfy the requirements of VDC requests. This means that there is no guarantee that the virtual data center can recover when a physical resource fails. A vast literature is dedicated to the concise approaches; see, e.g., [2, 3, 28, 30, 56, 70, 74, 77].

3.5 Objective function

Multiple VDC mappings can be created over the same physical data. Some mappings may be more efficient than others in terms of different objectives. In this work, the objectives that were pursued by the existing approaches to solve the VDCE problem with similar goals are classified into four objective function groups that are presented in the following subsections.

3.5.1 Economic profit-related objectives

Since the purpose of an InP is basically to maximize its profits, some studies have investigated the same goal of maximizing income, but proposed different modeling approaches. The important issue in these schemes is to achieve the cost saving while considering the QoS of cloud services and honoring the SLAs.

Maximizing the economic benefit could be achieved by:

- Minimizing operations costs related to the cost of using the data center that may include expenses for powering physical infrastructure, maintenance charges, etc [3, 70].
- Maximizing revenue of the infrastructure provider that can be done by ensuring the accommodation of the largest possible number of users' requests [3, 74].
- Avoiding penalties due to low service quality by means of satisfying the customer requirements by fulfilling their quality of service requirements [30].
- Maximizing utilization of resources to produce fully used servers and reduce wastage of resources [28].

3.5.2 Performance-related objectives

Certainly, the major concern of cloud client is to have a satisfactory performance on their running applications in the cloud. Accordingly, a good VDCE solution helps to serve as many customer requests as possible with the given set of resources, while guaranteeing QoS requirements defined by the service provider for each VDC request. This objective can be achieved by:

- Satisfying service-level objectives like response time, job completion time, etc. [28, 72].
- Minimizing reliability related to the system resilience to fault and its capacity of recovery [77].
- Balance load among infrastructure resources in order to minimize the stress on overutilized resources and distribute the load fairly between the others [56].

3.5.3 Energy-related objectives

Typically, cloud data centers are huge with too many processing, computing resources, power systems and cooling systems. Maintaining such large-scale infrastructures consumes enormous amount of energy and imposes a large carbon footprint on the environment [36, 73]. Due to the high cost of power consumption as well as the concerns for global warming and CO2 emissions, energy-aware VDCE schemes are very substantial in cloud computing. They attempt to generate more efficient and low power utilization data centers. This results in reduced maintenance costs for the InP. Thus, there are strong economic incentives to find an energy-efficient VDCE solution, such as minimizing the number of active server, links or other data center components.

- Minimize overall energy consumption since cloud data centers are huge with too many processing and computing resources, techniques for energy-efficient resource allocation must be addressed [30, 70].
- Minimize the number of running PMs by calculating the minimal number of active physical resources needed to host users' requests [72].
- Minimize carbon footprint to control the environmental impact of the infrastructures and ensure that carbon emissions generated by the leased resources should not exceed a fixed bound [2].

3.5.4 Network traffic objectives

The intra-data center and inter-data center network traffic has an important impact on the SLAs, revenue of the cloud providers and the efficiency of the cloud computing services. With the growing tendency toward communication applications, network-aware VDC assignment approaches have become very important. These approaches focus on the traffic connected subjects in the VDCE problem for the cloud data centers and attempt to decrease network traffic. In addition, some of these approaches distribute network traffic equally and attempt to prevent bottleneck. For instance, some applications use huge amounts of data that are to be mapped on specialized storage components, leading to considerable network traffic between compute servers and storage servers. In such cases, the available network bandwidth may become a bottleneck which has a significant effect on the SLAs, revenue of the cloud providers and the performance of the cloud computing services.

- Minimize network traffic to avoid situation of network congestion, meaning that the traffic load on some link exceeds its bandwidth capacity [3, 57].
- Minimize bandwidth consumption by keeping low utilization of each physical link [77].

Most of the previously mentioned objectives are not independent from each other or even conflicting; for instance, many objectives like energy consumption minimization could be transformed into economic profit objective. Hence, the VDCE strategy must balance between these goals.

3.6 Solution techniques

As mentioned previously, the VDCE problem is NP-hard. Based on this, we can classify the proposed techniques to solve the problem into two categories: exact algorithms and heuristics.

3.6.1 Exact algorithms

Since the VDCE problem holds the bin-packing problem as special case, which is NP-hard in the strong sense [49], there may be no chance for an exact algorithm with polynomial or pseudo-polynomial convergence time. Nevertheless, optimal techniques including integer linear programming (ILP) are proposed to solve VDCE. However, exact calculations usually attempt on taking care of little instances of the issue, so they will make benchmark results that represent an ideal bound to heuristic-based VDCE solutions. Software instruments implementing these algorithms, usually known as solvers, are accessible (e.g., CPLEX [33]). A good example is the work of [70].

3.6.2 Heuristics

Due to the fact that exact solutions require a high running time, it is necessary to use techniques with a balance between the optimal solution and runtime to embed the resources. Heuristic solutions would estimate strategies to solve optimization issues. Normally, heuristic solutions are created on the need of low duration of the time and employed on the complex instances of the issues. A vast literature has proposed this solution technique including: [2, 3, 28, 30, 56, 57, 70, 72, 74, 77].

3.6.3 Metaheuristics

As the exact solution for large-scale problems is hard to find, metaheuristics (like genetic algorithms [32], ant colony optimization [16], simulated annealing [38], etc.) can be used to find near-optimal solutions by improving a candidate solution with attention to a specific measure of quality. The following two approaches bring examples of metaheuristic-based solutions for VDCE: Gilesh et al. [23, 24].

4 State of the art in VDCE approaches

The VDCE problem can be seen in the literature in different forms; each one with the goal of developing the most effective embedding technique (placement or mapping in some sources). In this section, we provide a review of the literature that address the problem of resource allocation in virtualized data centers. For each work, we present a brief description of its context, its main goal and its solving techniques; then, we conclude it with a brief challenging critique.

We developed criteria for inclusion and exclusion of papers in order to narrow the scope of the analysis. To be included in the review, a study must: (a) have an exploration or examination of virtual resources allocation in both nodes and links in virtualized data centers; (b) be an empirical study or theoretical article; (c) be published in a peer-reviewed journal or by an international conference; (d) be written in English language and available for download from electronic database or library; and (e) be published during the current decade (from 2010 to 2018). We excluded from the review any literature that focused on issues related to virtual resource allocation with different use cases including: virtual network embedding, virtual machine placement and virtual cluster placement because these problems merit their own review. Table 1 illustrates the most significant characteristics of the reviewed research works. Table 2 shows the significations of acronyms given in Table 1.

Guo et al. [28] propose an architecture called SecondNet to implement a virtualized data center, which was proposed for the cloud to attain high scalability, elasticity, high utilization and practical deployment. Within this architecture, an algorithm for assigning servers and virtual links is the first attempt to solve the VDCE problem. The mapping algorithm considers the migration of servers in case of failure, or to improve network capacity data center, it can also increase or decrease the use of resources that were initially reserved for a request. The embedding algorithm consists of three main stages. In the first stage of mapping, the physical servers are grouped into clusters of different size and appropriate cluster is searched instead of the entire physical network when allocating the VDC request in order to minimize the allocation time instead of looking at each server for each. This further reduces communication costs because a VDC must be assigned to one group, characterized by the fact that the servers that make it up are located in a hop limit. In the second stage, virtual servers are assigned to the target group, and in the third stage, virtual links are assigned starting with the highest demand for bandwidth using the shortest path method. A robust spanning tree of the network topology has been obtained to signal the failures in the nodes or links to the VDC manager to adjust the affected VDCs. This ensures that the service is not affected for the victims of failures. If the path real location does not solve the problem of disconnection, it has to be immediately switched over to other solutions like virtual machine migration for uninterrupted services to the customers. SecondNet is simulated and verified the results in three network topologies like BCube, fat tree and VL2.

Zhani et al. [74] propose a dynamic model, called VDC planner that seeks to reduce energy costs. As [56], in addition to mapping servers and virtual links, also consider the switches as part of this process. Furthermore, it uses migration to both improve VDCE solution and better support the fluctuation of requests while decreasing total migration costs. VDC planner partitions the general problem into various usage scenarios, including VDCE, VDC scaling as well as dynamic VDC consolidation. The authors implemented two heuristic algorithms to support the previous scenarios. The first heuristic is created for migration-aware VDCE with the goal of finding a possible embedding of the request that incurs less migration cost. This problem is formulated as a minimum knapsack problem [10] and solved using simple greedy algorithm. The second heuristic is conceived for dynamic VDC consolidation. It attempts to reduce the number of running machines used to accommodate VDCs during light workload. The performance of the algorithm is evaluated by simulating the algorithm in a VL2 topology compared to a similar designed

Table 1 Comparison of virtual data center embedding approaches

References	Year	Objective				Link res.		Mapping meth.			Approach				Cloud arch.		Decomposition				Topology					
		EP	PR	ER	NT	BW	DY	EX	HR	MH	D	S	C	R	SG	MP	1S	2S	3S	2I	BC	DC	VL	ST	NS	FT
Guo et al. [28]	2010				●	●			●		●		●		●		●				●	●	●			●
Zhani et al. [74]	2013	●	●	●		●			●		●		●		●		●						●			
Rabbani et al. [56]	2013	●			●	●			●			●	●		●				●				●	●		
Amokrane et al. [3]	2013	●		●	●	●	●		●			●	●			●				●				●	●	
Rabbani et al. [57]	2014			●		●			●		●			●	●			●					●	●		
Zhang et al. [72]	2014	●	●	●		●			●			●	●		●		●						●			
Zuo et al. [77]	2014			●	●	●			●			●	●		●		●									●
Amokrane et al. [2]	2015	●		●	●	●	●		●		●		●			●				●				●		
Han et al. [30]	2015	●		●	●	●			●		●		●			●	●									●
Wen et al. [67]	2016	●			●	●			●			●	●		●		●									●
Sun et al. [63]	2016		●		●	●			●		●		●			●	●									●
Gilesh et al. [22]	2016	●				●		●	●		●		●		●			●								●
Yang et al. [70]	2017	●		●	●	●	●	●	●			●	●		●				●				●	●		●
Sun et al. [64]	2017				●	●			●			●	●			●	●									●
Ma [48]	2017	●				●			●			●	●			●	●									●
Cao et al. [9]	2017	●			●	●	●	●			●	●	●			●	●									
Nam et al. [54]	2017			●		●			●		●		●		●			●								●
Lo et al. [46]	2017				●	●		●	●		●			●	●			●								●
Yan et al. [69]	2017				●	●		●	●		●		●		●			●			●		●	●		●
Gilesh et al. [23, 24]	2018	●				●			●	●	●		●		●		●				●					●

Table 2 Acronyms' significations of the heads of Table 1

Head	Acronym	Signification
Objective	EP	Economic profit objective
	PR	Performance-related objective
	ER	Energy-related objective
	NT	Network traffic objective
Link resources	BW	Bandwidth
	DY	Delay
Mapping method	EX	Exact
	HR	Heuristic
	MH	Metaheuristic
Approach	D	Dynamic
	S	Static
	C	Concise
	R	Redundant
Cloud architecture	SG	Single
	MP	Multiple
Decomposition	1S	One stage
	2S	Two stages
	3S	Three stages
	2I	Inter-InP
Topology	BC	BCube
	DC	Dcell
	VL	VL2
	ST	Star
	NS	NSFNet
	FT	Fat tree

SecondNet [28] algorithm. The authors claim that their algorithm provides improvement in acceptance rate, revenue and the number of active machines. In this paper, single data center is considered for their algorithm simulation. The algorithm does not employ an effective embedding technique which creates modules to be allocated for each physical machine and does not account the virtual link between the virtual machines. Mapping of virtual links to physical links have not been incorporated to service the incoming request which enables the communication between the VMs a guaranteed service. VDC planner does not use availability information for VDCE. This information is required for reliable VDC request embedding which has to include the hardware failures and dependencies among various virtual components.

Rabbani et al. [56] seek to improve the acceptance of VDCs generating increased revenues supplier. Here, the authors introduce the process of assigning virtual switches and storage devices that had not been taken into account in previous proposals ([28] and [5]); it is also possible to define different kinds of resources for a single virtual device. The allocation algorithm consists of three stages. In the first phase, virtual servers are assigned to physical servers capable of supporting them, taking into account, as the main criterion, the proximity between them. In the second phase, virtual switches are assigned to the physical servers, taking into account the preallocation of virtual servers. Finally, in the third phase, they are assigned to virtual links between virtual nodes already assigned. Despite that the authors claimed that the algorithm works in diversified network topologies without any performance degradation, they have chosen VL2 and star topology for evaluating their algorithm. They also incorporated not only virtual machines, virtual links but also virtual switches for allocating the request. To attain efficiency in request embedding, the authors considered residual network bandwidth, load balancing, server virtualization and communication costs.

Amokrane et al. [3] present "Greenhead" as a proposal that considers several DCs geographically distributed through a central network. Its main objectives are to: (1) reduce the load on the core network (basically placing virtual machines, which consume large communication bandwidth in the same DC), (2) reduce emissions of carbon footprint (through the preferential use of renewable energy sources) and (3) maximize revenue provider (through efficient resource allocation that allows embedding most potential VDCs). It also ensures reduced energy costs given preference to regional energy rates and DCs preferring to make efficient use of it. The heuristic embedding algorithm consists of a central controller in charge of dividing each incoming VDC, monitor all parameters of DC, assign partitions to local DCs and assign links in the core network. It also consists of the local controllers that are responsible for mapping partitions locally, using different kinds of algorithms for resource embedding [5, 28, 74]. To evaluate Greenhead, an algorithm that does not divide reference VDC requests, but considers each virtual server as a partition, is simulated. The NSFNET topology is chosen for the core network that connects four data centers distributed in four different cities (with different climatic conditions and different prices for energy consumption). Under considerable number of variables, the algorithm obtains results as solution close to optimum with a reasonable computational time for small problem instances, improved utilization of the core network, increasing acceptance percentage VDCs, high energy use renewable and emission minimization carbon footprint per VDC.

Rabbani et al. [57] seek to ensure a high availability of the VDCs, while operating costs are minimized. Heterogeneity fault is considered on the network that involves a probability of different failure physical devices and a technique is proposed to calculate the availability of a VDC depending on the frequency of equipment failures and backup resources. The embedding algorithm is performed in two stages. In the first stage, virtual servers are assigned to physical resources, based on their status (active or inactive) and availability, seeking to reduce the number of active machines on the network and backup resources. In this process also, backup resources are provision to tack over any potential failure of the physical machines. In the next stage, the mapping of links and switches is done jointly, seeking to reduce consumed bandwidth. This VDCE model is tested on VL2 topologies (physical network) and star (requests VDC). An algorithm of reference is considered, in which virtual servers are scattered on physical servers to improve availability and backup resources are allocated randomly physical servers. Comparisons are made in terms of revenue, accepted VDC number, number of active machines, cost of backup resources and resource utilization by VDCs. The authors claim that the proposed algorithm gets better performance in all aspects.

Zhang et al. [72] propose an allocation system called Venice that is geared to provide reliable VDCs. Venice provides a technique to calculate the availability of a VDC based on failure characteristics of the physical devices and the dependencies between the elements that comprise it. Allocation algorithms in the two processes are distinguished: One realizes the allocation of VDC and the other the consolidation of VDCs. In the step of assigning each VDC, the incoming VDC request is being allocated to physical machines by examining the availability of resources. Initial embedding is performed with the VDC availability types. Venice availability-aware algorithm runs many times to achieve the best optimal embedding solution. The machine with lowest availability of resources will be dropped in each trial. The high availability of physical machine will be selected for embedding the current incoming VDC request. At the same time, over-provisioning of resources in the name of high availability is also to be considered to avoid high resource wastage. The next step reduces the number of active machines. Physical machines are sorted according to their order of current utilization. VM migration is performed from underutilized to highly utilized machines in such way to switch off underutilized machines to save energy. Here the replica of the node must not be embedded in the same physical machine in order to achieve fault tolerance for highly sensitive applications/data. Venice mainly builds architecture to adapt the frameworks with replicas which would be used in high risk and production data centers where reliability is the main concern. The reliability-aware embedding proceeds until high VDC availability is achieved to ensure the uninterrupted service to the consumers and also to provide replication of information to withstand the failures.

Zuo et al. [77] seek to guarantee the reliability requirement of the VDC, taking into account server failure only, while reducing the energy consumption. Since reducing bandwidth consumption helps reduce energy consumption, they focus on the trade-off between VDC bandwidth consumption and its reliability. Hence, they propose a heuristic VDCE algorithm called RAVDC. RAVDC is performed in three stages. The first stage aims at finding each VDC, the maximum number of VMs which can be embedded onto the same physical server. In the second stage of mapping, the VMs are grouped into groups of different sizes, where VMs with high communication service are grouped together, seeking to reduce communication costs. In the final stage, the embedding of server groups is done onto physical servers that can satisfy computing resource requirement starting with the powered ones, thus reducing energy consumption of the data center and then embed virtual links of VMs not embedded onto the same group. The performance of the algorithm is evaluated by simulating the algorithm in a VL2 topology compared to a reference energy-saving algorithm. Performance comparisons are made in terms of average reliability, average bandwidth consumption, average energy consumption and average embedding time. Simulation results show a clear improvement in bandwidth and energy consumption only. The proposed approach limits the reliability requirement in embedding two VMs belonging to the same VDC into the same physical host. Yet, reliability can be achieved using other techniques including: allocating back up resources to be used in case of hardware failure. Furthermore, link reliability is not considered in this study which could be also important for many cloud running applications.

Amokrane et al. [2] introduce a holistic framework named Greenslater that allows cloud providers to provide VDC requests across a geographically distributed infrastructure with the objective of reducing operational costs and SLA violation penalties, in addition to maximizing the utilization of the available renewable power, while considering the migration costs before migrating. Greenslater is a three-stage approach. The central controller first divides the newly arrived VDC request into partitions in a way that the intra-partition bandwidth is maximized and the inter-partition bandwidth is minimized. In the second step, the admission control decides whether to reject VDCs based on an estimation for carbon emission for the request. Next, using the partition embedding algorithm, all VDC partitions and connecting virtual links are mapped into the distributed infrastructure. To evaluate this framework, the authors considers physical infrastructure of four data centers, connected through the NSFNet topology as a backbone network. The authors perform a detailed evaluation of the sensitivity of their proposed solution to different arrival rates, location probability constraints, reporting periods and reconfiguration interval powers under both fixed and variable SLA violation penalty models. The simulation results prove that Greenslater achieves higher profit and ensures higher utilization of renewable and lower carbon footprint with minimum SLA violation. For instance, the proposed framework achieves profit gains of up to 33%, 53% and 129% compared to Greenhead, GreenheadNP and the load balancing approach, respectively.

Han et al. [30] introduce a holistic solution for resolving VDCE problem in multi-data center environment named SAVE with the goal of maximizing InPs revenue by saving the total energy consumption, while minimizing the service disruption. The main contribution of SAVE is that it provides the possibility to perform the inter-DC VM live migration. In addition, it expands the consolidated targets from VMs to network fabrics (e.g., paths and switches). The proposed algorithms are based on heuristic, consisting of two VDC embedded algorithms and a traffic engineering heuristic: (1) resource raising, for dealing with new VDCE request; (2) resource falling, for resource consolidation among hosts; and (3) traffic engineering, for realizing the network fabrics consolidation. The three algorithms are executed in sequential manner, which means that the traffic engineering goes after the resource raising/falling. To evaluate the proposed method, the authors exploited fat tree topology as a reference DCN topology and applied Gaussian distribution to resource demand model of VDCs. With the proposed resource raising heuristic, the number of required active hosts to accommodate the VDC requests was around two times less, compared to the result obtained from baseline approach (random allocation). Meanwhile, the number of active switches is 18.7% reduced after running the proposed traffic engineering heuristic. Those results are due to the fat tree topology which made it possible to pack all active servers under the same ToR switch, thus decreasing the number of active hosts drops. However, this proposal may not work fine with other data center network architectures those not share some features with fat tree topology, including BCube and Dcell [29].

Wen et al. [67] proposed a VDCE framework based on SDN for InP to manage the resources in a data center. The objective is to achieve higher revenue, while considering the trade-off among multiple requirements, such as the reliability and bandwidth occupation. This proposal uses two heuristic stages to embed a VDC: (1) First, VMs with more traffic are grouped, while VMs with less traffic are split into different partitions. Each partition will be mapped to a different physical resource. In this way, the communication between servers accommodating the VMs is minimized. (2) The second heuristic ensures the embedding of the partitions. Cluster of servers with enough resources and close to each other are located and assigned to host the VDC. This can reduce the hops of virtual links between VMs. The proposed approach is evaluated in terms of acceptance ratio, revenue, link occupation ratio and core link average utilization. The authors affirm that the proposal can achieve higher acceptance ratio, more revenue and less occupation rate of core links. Despite that, it fails to achieve better revenue-to-cost ratio compared to the existing algorithms. Thus, while the approach tries its best to allocate resources for VDCs, it causes more cost. Ideally, the mapping approach should ensure good performance reflected by acceptance ratio and revenue, yet minimize InP's expenses for powering physical infrastructure.

Sun et al. [63] consider the problem of reliable VDCE across multiple data centers. Similar to Wen et al. [67], the goal of the proposed approach is to minimize the total bandwidth consumption in backbone network for provisioning a VDC request, while satisfying its reliability requirement. The authors propose a three-step heuristic algorithm called RVDCE: (1) Group the physical servers according to reliability level. (2) Partition the VDC into many sub-VDCs, with the objective of reducing the traffic network between partitions, thereby minimizing the bandwidth consumption in the backbone network. (3) Embed the VDC partitions on data centers that satisfy the location constraints. VMs in the partition are embedded according to the following two steps: (1) embedding the VMs without backups and (2) embedding the VMs with backups. In order to evaluate the effectiveness and correctness of the proposed algorithm, two types of comparisons based on selection method of physical host are made: a case where VMs are mapped on the available servers with highest reliability (RVDCE_R) and another case where VMs are mapped onto servers with as low reliability as possible (RVDCE). The obtained results show that the proposed algorithm leads to a lower CPU and backbone bandwidth resource consumption compared to the existing approaches. Still, the authors only considered CPU allocation as the resource for the experiments which may not be sufficient to simulate real-world cloud resource allocation problem. Furthermore, the average backbone bandwidth consumption of RVDCE decreases with the growth of the size of partitions.

Gilesh [22] presents an algorithm for optimal placement of multiple virtual data centers on a physical data center. The objective is to maximize the number VDCs embedded, while minimizing the number of VM migrations. The author proposed a mixed approach with suitable two different modes—online mode or batch mode. Online embedding, based on local search heuristic, is applied whenever arrival of a new request or an exit of embedded request happens. It attempts to find a local solution with minimum modification to the existing mappings. Batch request embedding is formulated as a mixed integer program (MIP). It is attempted during the arrival of a set of requests or when it is hard to find a local solution. For simulation, the topologies were generated using the topology generator of FNSS toolchain [59]. He claims that the proposed multi-mode algorithm maximizes the resource utilization over time by the online embedding, without waiting for embedding in batches. The major limitation of the proposal is the slowness of ILP-based exact algorithm. The authors did not specify the number of physical servers used in their experimentation. However, a faster mathematical programming technique or near-optimal heuristics that can overcome the limitation is required to solve large-scale problems.

Yang et al. [70] addressed the VDCE problem with the goal of producing CSP's high long-term revenue, while minimizing energy consumption in light-workload data centers and minimizing embedding cost in heavy-workload data centers. In this paper, the authors proposed two different approaches. The first VDCE approach NSS-JointSL is performed in two stages: Once the VM mapping is done using nearest edge switch (NSS) approach, they propose a mapping algorithm that achieves the mapping goals by simultaneously embedding virtual switches and links and formulating as a mixed integer programming multi-commodity flow problem [65]. On the other hand, the second approach called Server-NSS-GBFS includes three stages: (1) mapping virtual servers as in the first approach; (2) mapping virtual switches in an energy-aware and random way; and (3) mapping virtual links by using a green breadth-first search (BFS) algorithm. The proposed approaches are evaluated under various network scales including: small-, medium-, large- and ultralarge-scale scenarios and multiple topologies. The performance of the proposed algorithms is compared with existing VDCE approaches of [57, 56]. Simulation results also demonstrate that NSS-JointSL is able to achieve the desired objectives, as well as that Server-NSS-GBFS is an effective and efficient VDCE algorithm. Although the proposed VDCE approaches outperformed previous works, the computation time of embedding a VDC request is too long, especially when more kinds of physical resources are considered. This issue is related to the usage of an exhaustive way for finding out feasible mapping solutions that resulted in severe degradation of performance with varying workload demands. Thus, the solution may not be appropriate for running applications on real-time systems.

Sun et al. [64] study the problem of embedding for evolving VDC request across federated data centers, such that the total operation cost and energy consumption are minimized. The aim of evolving VDCE is to embed the new arrived VDC requests to the physical infrastructure, assign physical resources for the new added virtual requests and minimize the amount of resource allocation or release the running physical components for the reduced or expired resource requirement. The proposed framework includes the following five scenarios: (1) initial embedding; (2) decrease resource requirements; (3) increase resource requirements; (4) inter-data center migration; and (5) periodic consolidation. The outcome of the experiments demonstrates that the proposed framework can reduce the resource and energy consumption much better than the existing solution that does not employ VM migration to adjust resource allocation of evolving VDC. Results proved that changing the embedding of links in backbone reduces backbone bandwidth resource consumption and results in lower embedding cost. Furthermore, it affirms that using migration strategy to achieve consolidation lowers the total embedding cost and energy consumption. While the proposed framework performs the resource allocation adjustment periodically, the challenge here is to detect the inadequate provision of resources and react quickly to avoid SLA violation and still optimal resource usage.

Ma [48] introduces a novel resource management framework based on SDN, allocating resources for VDC requests at the maximum revenue. Besides, considering different resource requirements, it is a heuristic embedding algorithm based on topological potential, called NMP. The major idea of NMP algorithm applies an iterative technique, which means generating a more exhaustive link bandwidth for the VMs between each other in a cluster when looking for the VM nodes in the cluster and mapping the cluster VM set to a host while meeting the reliability necessity of VDCs. Two different host selection methods are proposed in this paper: (1) The embedding is restricted to a cluster of hosts with more resources using the technique of node clustering and (2) hosts are scored according to its topological potential and the distance to the embedded VMs, so the host with the highest score is selected. In this paper, VDC requests are generated by the BRITE [51] topology generator. The data center network uses a fat tree topology with 6 ports, which includes 45 switches and 54 physical hosts. A large number of simulation experimental results show that the proposed algorithm can accept more requests at the minimum cost, and improve InP' revenues. However, while trying to achieve trade-off between reliability requirements and the embedding cost, the acceptance ratio and average revenue-to-cost ratio of VDCs of the proposed algorithm decrease with the increase in the VDC reliability requirements.

Cao et al. [9] propose a dynamic VDC provisioning algorithm inside metro-embedded DCs over metro optical networks to establish a data communications among the geographically distributed resources with high bandwidth, low latency and reduced energy consumption. They proposed an optimal VDC provisioning based on an integer linear programming (ILP) formulation to perform the initial embedding. The proposed formulation optimizes the total distance between users and DCs, considering the user resource requirement (e.g., compute resource) and available capacity in the DC. Considering the traffic variation and user dynamics, a dynamic VDC provisioning algorithm based on ILP is introduced to adapt the VDC. In order to reduce the complexity, the ILP is running only for the set of dynamic users. Experimental demonstrations conducted on the real-world implemented metro-embedded DC prototype test bed MEDC [11]. The results show that optimal and fast VDC mapping is realized based on the proposed ILP, while dynamic VDC reconfiguration makes sure that VDC is adjusted accordingly with the change in user location and resource requirement.

Nam et al. [54] focus on joint virtual data center embedding and server consolidation (HEA-E). The main objective of energy-aware methods in data centers is to reduce the total energy consumption of a physical DC. When a VDC request arrives, HEA-E first uses virtual machine mapping to map VMs onto physical servers. Subsequently, based on VM mapping results, virtual link mapping creates virtual links interconnecting newly mapped VM. In order to improve the resource efficiency, migration strategies are performed, namely: (1) partial migration that tries to consolidate some servers that are in underutilized situation, (2) migration on arrival that tries to reoptimize the whole system by remapping all existing requests every time when a new VDC request arrives in the system and (3) full migration that executes remapping of all existing requests upon a VDC joining or leaving. The performance of the proposed joint HEA-E embedding and migration strategies is compared with two VDCE algorithms, namely Greenhead [3] and SecondNet [28]. The authors claim that the new strategies can improve both resource utilization and energy efficiency of the data center, while keeping the complexity in an acceptable level. The proposed solution ignored the fact that network bandwidth usage by VM live migration could have negative effects on network efficiency for end users [62] because it uses significant bandwidth for a period of time. Accordingly, performing a consolidation operation on the whole DC upon each arrival of VM deployment and removal requests will significantly impacting applications QoS, causing too much interruption in the network, let alone extra computation complexity and delay. Thus, considering migration needs and overheads in the VDC consolidation algorithm could increase the overall data center performance and efficiency.

Lo et al. [46] focus on the survivability of VDC requests under any single switch failure, at one time. Since a switch failure results in multiple disconnected servers, to each primary VM, the algorithm allocates backup VM and ensures bandwidth reservation on backup paths. The approach called CALM attempt to minimize the total bandwidth consumption on every link for the active and reserved bandwidth. The proposed algorithm proceeds in two stages. VMP is performed first to place VMs into servers. Then, VLM is performed to allocate and reserve bandwidth for the communication demands between VMs even after failures. The simulations were made on a fat tree data center topology. The authors evaluated the network resource usage of the proposed solution and show that CALM uses only additional 13% network resource to guarantee survivability as compared to a typical VDC strategy. In order to address the cascade of multiple failures, the algorithm must be applied multiple times to calculate a new resource allocation. However, the process needs to reserve a huge amount of cloud resources that may lead to low resource utilization. Therefore, it is crucial to extend the algorithm to handle multiple failures with cost-effective resource allocation.

Yan et al. [69] investigate the embedding problem for heterogeneous bandwidth VDC (star topology) in physical DCs of general topology while minimizing the network traffic between servers and switches. The goal is to reduce VDC request rejections caused by bottleneck physical links. The embedding problem was decomposed into two coupled subproblems: virtual machine (VM) placement and multi-path link mapping. First, given a certain VM placement, they formulated the link assignment problem with linear programming to minimize the maximum link utilization. Next, they proposed a heuristic algorithm, referred to as the perturbation algorithm, for the VM placement. The algorithm will be triggered if the first-fit method fails to place a VM due to network congestion. The perturbation method modifies the current placement by removing one VM each time from the bottleneck server that contributes the most traffic to the hot links. Simulation was made on small-sized standard DC architectures, including fat tree, VL2 and BCube where each consists of 16 servers. Comparisons were made in terms of success rate of VDCE, and the results show that the algorithm performs better in comparison with the existing well-known algorithms: first-fit, next-fit and greedy algorithms. However, while quick response is a crucial metric for ensuring a high service quality of the VDC initial embedding, the paper did not consider the computation time required by the algorithm to solve the embedding. We believe that the proposal will fail to achieve low solving time as it uses linear programming to solve link mapping in each VM remapping iteration of perturbation algorithm. Hence, solution could not be applied for medium- and large-scale scenarios.

Gilesh et al. [23, 24] address the problem of finding the cost-effective set of VM migrations, to optimally embed an incoming VDC request across the fragmented resources. The objective is to minimize the violation of SLAs. In this paper, a greedy algorithm and two metaheuristic solutions are proposed for the problem of finding virtual machine migrations with minimum cost based on artificial bee colony and simulated annealing optimization. The algorithms used as input data, the current mapping of VMs, the arrived VDC request and the list of VMs and PMs with their configurations. The authors claim that experimental results show that the proposed technique can reduce the average migration time up to 30% and the penalty for corresponding service-level agreement violation by 200%. The major advantage is that unlike [69], the proposed technique is appropriate for mapping applications having critical performance requirements. The usage of metaheuristic solution provides a near-optimal solution within a reasonable time which can much reduce scheduling waiting time.

5 Discussion

Several studies have contributed to find the efficient placement of the VDC requests by applying the embedding schemes as presented in the previous sections. This section provides a complete comparative analysis of the VDCE schemes proposed for the cloud environment and the issues which need to be addressed.

5.1 Multi-objective

Most of the current works in embedding virtual data centers consider only one optimization objective or combine multiple objectives into a composite one with weighting. However, these may be conflicting objectives when considered altogether. For example, packing VMs onto a reduced number of PMs and shutting off others is an effective way to minimize energy costs. However, concentrating workloads on a minimal subset of the system resources can cause heat imbalances and create hot spots, which may impact cooling costs and degrade server life and performance. An effective strategy should consider trade-offs among all these objectives.

5.1.1 Power-related factors in VDCE approaches

Some power-aware VDCE schemes are suggested for cloud computing environment. Table 3 displays the aspects which power-aware VDCE algorithms consider in their placement decisions. The significations of the table headers are as follows:

Table 3 Comparison of power-related VDCE approaches

	PM states	PM power usage	Network element power usage	Carbon emission
Zhani et al. [74]		●		
Amokrane et al. [3]		●	●	●
Rabbani et al. [57]		●	●	
Zhang et al. [72]		●		
Zuo et al. [77]	●			
Amokrane et al. [2]		●	●	●
Yang et al. [70]		●	●	
Han et al. [30]		●	●	
Nam et al. [54]		●	●	

- PM states: Calculating the minimal number of active physical resources needed to host users' requests.
- PM power usage: Decreasing the amount of power that various server components (CPU, memory, etc.) consume.
- Network element power usage: Minimizing the power used by network routing elements.
- Carbon emission: Ensuring that carbon emissions generated by the leased resources do not exceed a fixed bound.

Table 1 reveals that most of the popular cloud data centers utilize different categories of PMs. These PMs differ in their resource capacity: Some machines are larger than others, whereas some others have relatively higher capacity of one type of resource compared to other resources. Thus, it is obvious that the amount of consumed energy depends on the resource capacity of the running PM. That is why most approaches consider power usage cost for each PM rather than PM state factor. However, other approaches consider uniform data centers with identical servers [77]. In this case, the overall energy consumption depends only on the number of the running PMs, while the consumed energy for each machine is fixed. For this reason, it is more convenient to apply the server state factor.

5.1.2 Network-related factors in VDCE approaches

A strongly related issue in the mapping of VDCs on infrastructure resources is the network loads. For this purpose, some studies try to reduce network traffic while others distribute network traffic evenly and try to avoid congestion. Table 4 indicates the factors which network-aware VDCE approaches apply to produce mappings that will result in less communication in the data centers. The significations of the table headers are as follows:

- Traffic between VMs: Decreasing the congestion of network between the VMs.

Table 4 Comparison of network-related VDCE approaches

	Traffic between VMs	Traffic between remote cloud sites	Traffic between PMs	Communication distance between VMs
Guo et al. [28]	•			
Rabbani et al. [56]			•	•
Amokrane et al. [3]		•		
Zuo et al. [77]				•
Amokrane et al. [2]		•		
Yang et al. [70]	•			
Han et al. [30]	•			
Sun et al. [63]	•	•		
Sun et al. [64]	•	•		
Wen et al. [67]				•
Yan et al. [69]	•		•	
Lo et al. [46]	•			
Cao et al. [9]				•

- Traffic between remote cloud sites: Reducing traffic in the backbone clouds network.
- Traffic between PMs: Distributing network traffic evenly in order to avoid bottleneck of physical links.
- Communication distance between VMs: Minimizing the hop count between the virtual nodes (i.e., VM or virtual switch).

Typically, the significance of physical links differs in VDCE. For instance, the core links are more important than the edge links in data center since a core link may be used by more VMs than an edge link mainly. A virtual link can be mapped to several physical links with different weights. The objective of minimizing traffic between the VMs is considered by many approaches, since it attempts to respect the network cost associated with each physical link accommodating the virtual path between VMs. However, analogous to uniform servers in data centers, some approaches treat all the physical links evenly without discrimination [67, 77]. Here, the problem can be reduced to minimize the count of physical links forming the virtual path between VMs or the number of the intermediate physical nodes.

5.1.3 Profit-related VDCE approaches

Profit-aware VDCE approaches try to diminish the data center maintenance costs for the cloud providers. Table 5 introduces the factors applied in the profit-aware VDC embedding approaches to generate a better placement. The significations of the table headers are as follows:

Table 5 Comparison of profit-related VDCE approaches

	Node migration	VDC revenue	Resource usage
Zhani et al. [74]	•	•	
Rabbani et al. [56]			•
Amokrane et al. [3]		•	
Zhang et al. [72]	•		
Amokrane et al. [2]	•	•	
Yang et al. [70]		•	
Han et al. [30]	•		
Ma [48]		•	
Wen et al. [67]		•	
Gilesh et al. [22]	•		
Gilesh et al. [23, 24]	•		

- Node migration: Decreasing the cost due to migrating virtual node from a physical node to another.
- VDC revenue: Maximizing the total revenue generated from the embedded VDC requests.
- Resources usage: Reducing server fragmentation.

For profit-aware VDCE approaches, improving the revenue earned by the InP is the most considered. The objective of the InP is to maximize the difference between the revenue and the costs. The revenue is proportional to a weighted sum of the total resources (CPU, memory, disk, bandwidth) used by a VDC during a time slot. However, cost refers to the amount of resources the VDC occupies as well as penalties due to migration, which is also considered by some approaches.

5.1.4 Performance-related VDCE approaches

Performance-aware VDCE approaches try to provide a better QoS for the cloud clients. Table 6 indicates the factors which performance-aware VDCE approaches apply. The significations of the table headers are as follows:

- Waiting delay: Minimizing request scheduling delay, which refers to the time a request spends in the waiting queue before it is scheduled.
- Fault restoration: Minimizing the failure recovery costs of a node or a link.

Table 6 Comparison of performance-related VDCE approaches

	Waiting delay	Fault restoration	Unavailability
Zhani et al. [74]	•		
Zhang et al. [72]		•	•
Zuo et al. [77]			•
Sun et al. [63]		•	•

- Unavailability: Decreasing the penalty from failing to meet the reliability of a VDC.

Provision of QoS guarantees to high-performance applications can be critical to the success of the services provided to their users. For end users, execution time or response time of an application is usually the first choice for QoS specification. A mechanism for ensuring service quality in terms of execution deadlines in a cloud computing environment is a desirable goal. That is why most performance-related approaches consider minimizing unavailability problems to provide better performance.

A multi-objective characteristic is inherent in cloud resource allocation, as the goals of cloud providers, cloud users and other stakeholders can be independent. For instance, cloud provider may attempt to provide optimal mapping in order to reduce resources costs, while cloud users are interested in QoS. Even among cloud users themselves, different users or the same user at different times may require different QoS, such as low computational costs and high speeds. Some multi-objective VM placement approaches have already been proposed [4, 45]. Hence, a multi-objective nature of the VDCE problem could become more and more significant in the future of cloud computing. However, taking into account all objectives seems to be impossible due to conflicts of interest between clients.

5.2 Cloud elasticity

Cloud elasticity [20] is an important issue in cloud resource management process. The concept of elasticity is defined as the degree to which a system is able to react to workload changes by provisioning and de-provisioning resources in an autonomic manner, such that at each point in time the available resources match the current demand as closely as possible [31]. Table 1 shows that most works have addressed the problem of dynamic VDCE. Still, through surveying the VDCE approaches, we concluded that only [64] has focused on the issues related to elastic cloud. There are three basic methods employed in the implementation of elasticity solutions: (1) replication which consists of adding/removing instances from user virtual environment (VMs, disk attached to VM, etc.); (2) resizing methods where processing, memory and storage resources can be added/removed from a running virtual instance; and (3) VM migration can be adopted while handling insufficient resources in cloud. To accommodate the resources, VMs can be migrated between hosts. The challenge here is to detect the inadequate provision of resources and react quickly to avoid SLA violation and still optimal resource usage.

5.3 Survivability

A failure in the physical data center can affect several virtual resources. Therefore, the survivability has to be considered in the embedding of the virtual resources. Node and link failure survivability problems have been investigated extensively for optical and multi-protocol label-switched (MPLS) networks [42] and real-time systems [75]. In the presence of arbitrary resource failures, the two well-known approaches for handling failures are: protection and restoration. Protection is done in a proactive way and usually employed during the embedding phase by allocating backup resources before any failure happens. Table 1 reveals that only two approaches [46, 57] have considered the protection technique while embedding VDCs. On the other hand, restoration reacts after the failure occurs and starts the backup restoring mechanism and it is more reactive in nature. Among the surveyed approaches, only the work of [72] has deployed a mechanism for failure restoration. The key to efficient restoration mechanisms is survivable mapping in the presence of link failures. However, those works focus on single physical failure. Types of failures are single link, single facility node and single regional failures in the network. They assume that the network failures are independent from each other and only one failure happens at a time.

5.4 Optimization algorithms

VDCE problem is considered as a non-deterministic polynomial-time hard (NP-hard) problem. Generally, it is difficult to develop algorithms for producing optimal solutions within a short period of time for this type of problem. Metaheuristic techniques can deal with these problems by providing near-optimal solutions within a reasonable time. Metaheuristics have become popular in the past years due to their efficiency to solve large and complex problems. However, Table 1 shows that only one work [23, 24] has implemented a metaheuristic algorithm to solve the VDCE problem namely: an artificial bee colony and simulated annealing algorithms. A number of metaheuristic algorithms have been used to solve the virtual machine placement problems [1], namely simulated annealing (SA), genetic algorithm (GA), ant colony optimization (ACO), particle swarm optimization (PSO) and biogeography-based optimization (BBO). However, there are only two metaheuristic algorithms that have been used in the VDCE context Gilesh et al. [23, 24].

5.5 Green computing

Due to the high cost of power consumption as well as the concerns for global warming and CO2 emissions, recently, the green data centers have become an important issue for cloud service providers and InPs are facing a lot of pressure to make their infrastructure greener and environment friendly. The concept of green computing refers to the attempts to maximize the use of power consumption and energy efficiency and to minimize the cost and CO2 emission [44]. Development of new computing models, computer systems and applications having low cost and low energy consumption is the primary purposes of green computing [26]. Table 1 reports that numerous works has tackled the problem of maximizing energy efficiency. However, Table 3 shows that only the works of [2, 3] have examined the problem of green VDCE. More efforts are needed to study the relationship between varying workloads, while an attempt should be made to build frameworks for data centers that operate on renewable sources of energy (e.g., solar and wind power).

5.6 Distributed system

In a centralized VDCE system, one central entity is responsible for performing the embedding. It has a global view of the data center and takes decisions according to the descriptions of the available physical resources. As far as we know, all the VDCE solutions presented in the literature are centralized. The advantage of such approach is that the assignment is performed, while the entity is aware of the global data center situation, which makes the embedding more optimal. However, it occurs scalability problems in large-scale data centers and presents a single point of failure (if the central entity fails, the entire mapping process fails). On the contrary, in a distributed VDCE system, multiple entities compute the embeddings. Some distributed VN embedding approaches are proposed [7, 17]. The main advantage of such approach is scalability, but communication cost and synchronization overhead need to be reduced.

5.7 Co-location interference

When determining to place a set of VMs on a PM, many works only investigate that the total requested resources of the VMs do not exceed the PM's capacity. However, in practice, there are also other features of the VMs that influence how appropriate they are for co-location. One feature to respect is correlation, which takes into account the resource demand of several of the VMs will increase at the same time [66]. Another feature is the noisy neighbor effect: Since present virtualization technologies do not allow entire performance isolation of the colocated VMs, if one of the VMs consumes a resource excessively, this may degrade the performance of the others [37]. The interested reader can refer to the works in [39–41] that tackle these issues in cloud and high-performance computing areas.

5.8 Evaluation environment

Even though several algorithms are presented to solve the VDCE problem, they are mostly not comparable because (1) either does not exist a common evaluation framework or (2) some minor technical issues forbid direct comparison. Unlike VNE problem, network topology generators are used to generate physical and virtual infrastructures including GM-ITM [71] and BRITE [51]. Thus, creating or extracting from any of the papers a data set/methodology to actually compare future algorithms would be very beneficial for the community. Especially as the VDC is a quite concise problem, collecting a reasonable data set might be easier as for the related problems.

6 Conclusion and future directions

Infrastructure as a service (IaaS) cloud computing providers, with flexible on-demand infrastructures and pay-as-you-go pricing models, have become an alternative to the rising total cost of ownership for computing infrastructures in many enterprises. The rapidly growing customer demands for cloud resources have increased the role of data centers in recent years. The scientific community has shown that data center virtualization is an alternative to solve the problems they are currently experiencing. Recently, the VDCE problem has become a hot topic in the cloud data centers because proper placement can make great contributions to provide a high-quality cloud value-added service to end users.

In this paper, we have surveyed the past and the state-of-the-art VDCE approaches proposed for the cloud computing data centers. For this reason, first we provide the definition and objectives, and then, we present taxonomy of the VDCE approaches based on the type of the placement algorithm. Then, we describe various proposed VDCE approaches, and the VDCE factors in each approach are examined to clarify the advantages, properties and limitations of each mapping algorithm in detail. Additionally, we present a complete comparative study of the VDCE approaches which feature the items that should be studied in the future researches. It is evident that VDCE provides a promising virtual resource allocation and scheduling methodology about physical resource utilization, load balancing, green computing, overload avoidance, traffic fluctuation and QoS in cloud computing environment. However, for this to become reality, some issues must be addressed in the next years, in order to eliminate all obstacles that would prohibit the wide use of cloud computing. As specified in our comparisons, most VDCE approaches are designed to improve the performance and energy-related problems in the cloud data centers and have ignored the survivability-related objectives in the VDCE algorithms. On the other hand, with the ever-increasing data center size growth, hardware failures are inevitable, which may lead to a degradation in the service performance. Moreover, security is one of the crucial factors which should be considered in the future VDCE researches and studies, and its negligence makes the whole cloud system vulnerable to various security attacks and hinders the cloud computing critical role in the future IT systems.

Acknowledgements We dedicate this research work for the memory of our deceased co-author Prof. Maher Ben Jemaa. We thank the editor and the eighteen anonymous referees who have provided valuable comments on an earlier version of this paper. We would also like to show our gratitude to Mrs Jabeen Nazeer Hussain (Faculty member at the College of Computer Engineering and Sciences, Prince Sattam Bin Abdulaziz University) for the English proofreading of the last version of this paper.

References

1. Alboaneen DA, Tianfield H, Zhang Y (2016) Metaheuristic approaches to virtual machine placement in cloud computing: a review. In: 15th International Symposium on Parallel and Distributed Computing (ISPDC). IEEE, pp 214–221
2. Amokrane A, Langar R, Zhani MF, Boutaba R, Pujolle G (2015) Greenslater: on satisfying green slas in distributed clouds. IEEE Trans Netw Serv Manag 12(3):363–376
3. Amokrane A, Zhani MF, Langar R, Boutaba R, Pujolle G (2013) Greenhead: virtual data center embedding across distributed infrastructures. IEEE Trans Cloud Comput 1(1):36–49
4. Ashraf A, Porres I (2018) Multi-objective dynamic virtual machine consolidation in the cloud using ant colony system. Int J Parallel Emerg Distrib Syst 33(1):103–120
5. Ballani H, Costa P, Karagiannis T, Rowstron A (2011) Towards predictable datacenter networks. In: ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), vol 41. ACM, pp 242–253

6. Bari MF, Boutaba R, Esteves R, Granville LZ, Podlesny M, Rabbani MG, Zhang Q, Zhani MF (2013) Data center network virtualization: a survey. IEEE Commun Surv Tutor 15(2):909–928
7. Beck MT, Fischer A, de Meer H, Botero JF, Hesselbach X (2013) A distributed, parallel, and generic virtual network embedding framework. In: 12th IEEE International Conference on Communications (ICC). IEEE, pp 3471–3475
8. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gen Comput Syst 28(5):755–768
9. Cao X, Popescu I, Chen G, Guo H, Yoshikane N, Tsuritani T, Wu J, Morita I (2017) Optimal and dynamic virtual datacenter provisioning over metro-embedded datacenters with holistic SDN orchestration. Opt Switch Netw 24:1–11
10. Carnes T, Shmoys D (2008) Primal-dual schema for capacitated covering problems. In: International Conference on Integer Programming and Combinatorial Optimization. Springer, pp 288–302
11. Chen G, Guo H, Zhang D, Zhu Y, Wang C, Yu H, Wang Y, Wang J, Wu J, Cao X et al (2015) First demonstration of holistically-organized metro-embedded cloud platform with all-optical interconnections for virtual datacenter provisioning. In: 2015 Opto-Electronics and Communications Conference (OECC). IEEE, pp 1–3
12. Chowdhury NMK, Boutaba R (2010) A survey of network virtualization. Comput Netw 54(5):862–876
13. Correa ES, Fletscher LA, Botero JF (2015) Virtual data center embedding: a survey. IEEE Latin Am Trans 13(5):1661–1670
14. Costa MC, Létocart L, Roupin F (2005) Minimal multicut and maximal integer multiflow: a survey. Eur J Oper Res 162(1):55–69
15. Dally WJ, Towles BP (2004) Principles and practices of interconnection networks. Elsevier, Amsterdam
16. Dorigo M (1992) Optimization, learning and natural algorithms. PhD Thesis, Politecnico di Milano
17. Esposito F, Di Paola D, Matta I (2016) On distributed virtual network embedding with guarantees. IEEE/ACM Trans Netw 24(1):569–582
18. Fischer A, Botero JF, Beck MT, De Meer H, Hesselbach X (2013) Virtual network embedding: a survey. IEEE Commun Surv Tutor 15(4):1888–1906
19. Fox A, Griffith R, Joseph A, Katz R, Konwinski A, Lee G, Patterson D, Rabkin A, Stoica I (2009) Above the clouds: a Berkeley view of cloud computing. Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS 28:13
20. Galante G, de Bona LCE (2012) A survey on cloud computing elasticity. In: 5th IEEE International Conference on Utility and Cloud Computing (UCC). IEEE, pp 263–270
21. Ghomi EJ, Rahmani AM, Qader NN (2017) Load-balancing algorithms in cloud computing: a survey. J Netw Comput Appl 88:50–71
22. Gilesh M (2016) Towards a complete framework for virtual data center embedding. arXiv preprint arXiv:1611.06309
23. Gilesh M, Kumar S, Jacob L (2018) Bounding the cost of virtual machine migrations for resource allocation in cloud data centers. In: Proceedings of the 33rd Annual ACM Symposium on Applied Computing. ACM, pp 201–206
24. Gilesh M, Satheesh S, Kumar S, Jacob L (2018) Selecting suitable virtual machine migrations for optimal provisioning of virtual data centers. ACM SIGAPP Appl Comput Rev 18(2):22–32
25. Greenberg A, Hamilton JR, Jain N, Kandula S, Kim C, Lahiri P, Maltz DA, Patel P, Sengupta S (2009) Vl2: a scalable and flexible data center network. In: ACM SIGCOMM Computer Communication Review, vol 39. ACM, pp 51–62
26. Guo B, Shen Y, Shao Z (2009) The redefinition and some discussion of green computing. Chin J Comput 32(12):2311–2319
27. Guo C, Lu G, Li D, Wu H, Zhang X, Shi Y, Tian C, Zhang Y, Lu S (2009) Bcube: a high performance, server-centric network architecture for modular data centers. ACM SIGCOMM Comput Commun Rev 39(4):63–74
28. Guo C, Lu G, Wang HJ, Yang S, Kong C, Sun P, Wu W, Zhang Y (2010) Secondnet: a data center network virtualization architecture with bandwidth guarantees. In: 6th International Conference on emerging Networking EXperiments and Technologies (CoNEXT). ACM, p 15
29. Guo C, Wu H, Tan K, Shi L, Zhang Y, Lu S (2008) Dcell: a scalable and fault-tolerant network structure for data centers. In: ACM SIGCOMM Computer Communication Review, vol 38. ACM, pp 75–86

30. Han Y, Li J, Chung JY, Yoo JH, Hong JWK (2015) Save: energy-aware virtual data center embedding and traffic engineering using SDN. In: 1st IEEE Conference on Network Softwarization (NetSoft). IEEE, pp 1–9
31. Herbst NR, Kounev S, Reussner RH (2013) Elasticity in cloud computing: what it is, and what it is not. In: 10th International Conference on Autonomic Computing (ICAC), vol 13, pp 23–27
32. Holland JH et al (1992) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press, Cambridge
33. ILOG I (2008) Ilog cplex: high-performance software for mathematical programming and optimization. http://www.ilog.com/products/cplex
34. Infrastructure, C.D.C.: 2.5 design guide (2010)
35. Jennings B, Stadler R (2015) Resource management in clouds: survey and research challenges. J Netw Syst Manag 23(3):567–619
36. Jing SY, Ali S, She K, Zhong Y (2013) State-of-the-art research study for green cloud computing. J Supercomput 65(1):445–468
37. Kim S, Eom H, Yeom HY (2013) Virtual machine consolidation based on interference modeling. J Supercomput 66(3):1489–1506
38. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680
39. Kocoloski B, Lange J (2015) Xemem: efficient shared memory for composed applications on multi-os/r exascale systems. In: 24th International Symposium on High-Performance Parallel and Distributed Computing. ACM, pp 89–100
40. Kocoloski B, Lange J (2016) Lightweight memory management for high performance applications in consolidated environments. IEEE Trans Parallel Distrib Syst 27(2):468–480
41. Kocoloski B, Zhou Y, Childers B, Lange J (2015) Implications of memory interference for composed HPC applications. In: International Symposium on Memory Systems. ACM, pp 95–97
42. Lau W, Jha S (2004) Failure-oriented path restoration algorithm for survivable networks. IEEE Trans Netw Serv Manag 1(1):11–20
43. Leiserson CE (1985) Fat-trees: universal networks for hardware-efficient supercomputing. IEEE Trans Comput 100(10):892–901
44. Li Q, Zhou M (2011) The survey and future evolution of green computing. In: IEEE/ACM International Conference on Green Computing and Communications. IEEE Computer Society, pp 230–233
45. Liu F, Liu Z et al (2012) Multi-objective optimization for initial virtual machine placement in cloud data center. J Inf Comput Sci 9(16):5029–5038
46. Lo HY, Liao W (2017) Calm: survivable virtual data center allocation in cloud networks. IEEE Trans Serv Comput. https://doi.org/10.1109/TSC.2017.2777979
47. Lodi A, Martello S, Monaci M (2002) Two-dimensional packing problems: a survey. Eur J Oper Res 141(2):241–252
48. Ma J (2017) Resource management framework for virtual data center embedding based on software defined networking. Comput Electric Eng 60:76–89
49. Martello S, Toth P (1990) Knapsack problems: algorithms and computer implementations. Wiley, New York
50. Masdari M, Nabavi SS, Ahmadi V (2016) An overview of virtual machine placement schemes in cloud computing. J Netw Comput Appl 66:106–127
51. Medina A, Lakhina A, Matta I, Byers J (2001) Brite: an approach to universal topology generation. In: 9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS). IEEE, pp 346–353
52. Meng X, Pappas V, Zhang L (2010) Improving the scalability of data center networks with traffic-aware virtual machine placement. In: 2010 Proceedings IEEE INFOCOM. IEEE, pp 1–9
53. Mishra M, Das A, Kulkarni P, Sahoo A (2012) Dynamic resource management using virtual machine migrations. IEEE Commun Mag 50(9):34–40
54. Nam TM, Thanh NH, Hieu HT, Manh NT, Van Huynh N, Tuan HD (2017) Joint network embedding and server consolidation for energy-efficient dynamic data center virtualization. Comput Netw 125:76–89
55. Quinn P, Nadeau T (2015) Problem statement for service function chaining. Tech. rep
56. Rabbani MG, Pereira Esteves R, Podlesny M, Simon G, Zambenedetti Granville L, Boutaba R (2013) On tackling virtual data center embedding problem. In: IFIP/IEEE International Symposium on Integrated Network Management. IEEE, pp 177–184

57. Rabbani MG, Zhani MF, Boutaba R (2014) On achieving high survivability in virtualized data centers. IEICE Trans Commun 97(1):10–18
58. Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA (2012) Heterogeneity and dynamicity of clouds at scale: Google trace analysis. In: 3rd ACM Symposium on Cloud Computing. ACM, p 7
59. Saino L, Cocora C, Pavlou G (2013) A toolchain for simplifying network simulation setup. SimuTools 13:82–91
60. Schneider F, Egawa T, Schaller S, Hayano SI, Schöller M, Zdarsky F (2014) Standardizations of SDN and its practical implementation. NEC Technical Journal, Special Issue on SDN and Its Impact on Advanced ICT Systems 8.2
61. Sivaranjani B, Vijayakumar P (2015) A technical survey on various VDC request embedding techniques in virtual data center. In: 2015 National Conference on Parallel Computing Technologies (PARCOMPTECH), pp 1–6
62. Stage A, Setzer T (2009) Network-aware migration control and scheduling of differentiated virtual machine workloads. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing. IEEE Computer Society, pp 9–14
63. Sun G, Bu S, Anand V, Chang V, Liao D (2016) Reliable virtual data center embedding across multiple data centers. In: 1st Conference on Internet of Things and Big Data, pp 1059–1064
64. Sun G, Liao D, Bu S, Yu H, Sun Z, Chang V (2017) The efficient framework and algorithm for provisioning evolving VDC in federated data centers. Future Gen Comput Syst 73:79–89
65. Szeto W, Iraqi Y, Boutaba R (2003) A multi-commodity flow based approach to virtual network resource allocation. In: IEEE Global Telecommunications Conference (GLOBECOM), vol 6. IEEE, pp 3004–3008
66. Verma A, Dasgupta G, Nayak TK, De P, Kothari R (2009) Server workload analysis for power minimization using consolidation. In: Conference on USENIX Annual Technical Conference. USENIX Association, pp 28–28
67. Wen X, Han Y, Yu B, Chen X, Xu Z (2016) Towards reliable virtual data center embedding in software defined networking. In: IEEE Military Communications Conference (MILCOM). IEEE, pp 1059–1064
68. Xie D, Ding N, Hu YC, Kompella R (2012) The only constant is change: incorporating time-varying network reservations in data centers. ACM SIGCOMM Comput Commun Rev 42(4):199–210
69. Yan F, Lee TT, Hu W (2017) Congestion-aware embedding of heterogeneous bandwidth virtual data centers with hose model abstraction. IEEE/ACM Trans Netw (TON) 25(2):806–819
70. Yang Y, Chang X, Liu J, Li L (2017) Towards robust green virtual cloud data center provisioning. IEEE Trans Cloud Comput 5(2):168–181
71. Zegura EW, Calvert KL, Bhattacharjee S (1996) How to model an internetwork. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), vol 2, pp 594–602
72. Zhang Q, Zhani MF, Jabri M, Boutaba R (2014) Venice: Reliable virtual data center embedding in clouds. In: IEEE Conference on Computer Communications (INFOCOM). IEEE, pp 289–297
73. Zhang Z, Su S, Lin Y, Cheng X, Shuang K, Xu P (2015) Adaptive multi-objective artificial immune system based virtual network embedding. J Netw Comput Appl 53:140–155
74. Zhani MF, Zhang Q, Simon G, Boutaba R (2013) VDC planner: dynamic migration-aware virtual data center embedding for clouds. In: IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE, pp 18–25
75. Zheng Q, Shin KG (1998) Fault-tolerant real-time communication in distributed computing systems. IEEE Trans Parallel Distrib Syst 9(5):470–480
76. Zou J, Yan F, Lee TT, Hu W (2014) A perturbation algorithm for embedding virtual data centers in multipath networks. In: IEEE Global Communications Conference (GLOBECOM). IEEE, pp 2240–2245
77. Zuo C, Yu H, Anand V (2014) Reliability-aware virtual data center embedding. In: 6th International Workshop on Reliable Networks Design and Modeling (RNDM). IEEE, pp 151–157