



# Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities

Mohammad Masdari<sup>1</sup> · Mehran Zangakani<sup>2</sup>

Published online: 23 October 2019

© Springer Science+Business Media, LLC, part of Springer Nature 2019

## Abstract

Efficient task and workflow scheduling are very crucial for increasing performance, resource utilization, customer satisfaction, and return of investment for cloud service providers. Based on the number of clouds that the scheduling schemes can support, they can be classified as single-cloud and inter-cloud scheduling schemes. This paper presents a comprehensive survey and an overview of the inter-cloud scheduling schemes aimed to allocate user-submitted tasks and workflows to the appropriate virtual machines on multiple clouds regarding various objectives and factors. It classifies the scheduling schemes designed for a variety of inter-cloud environments and describes their architecture, key features, and advantages. Also, the inter-cloud scheduling approaches are compared and their various features are highlighted. Finally, the concluding remarks and open research challenges in the multi-cloud scheduling context are illuminated.

**Keywords** Task · Workflow · Scheduling · Virtual machine · Multi-cloud · Federation · Distributed cloud

## 1 Introduction

Cloud computing is a network-based growing technology aimed to provide various IT services according to the pay-for-use model for a variety of organizations and customers [1]. It mainly applies virtualization to share the data centers (DCs) resources among multiple VMs and conduct more effective and energy-efficient resource management. However, the increasing demand for more resources in

---

✉ Mohammad Masdari  
com\_eng1@yahoo.com; M.Masdari@iaurmia.ac.ir

Mehran Zangakani  
Mehranzangakany@gmail.com

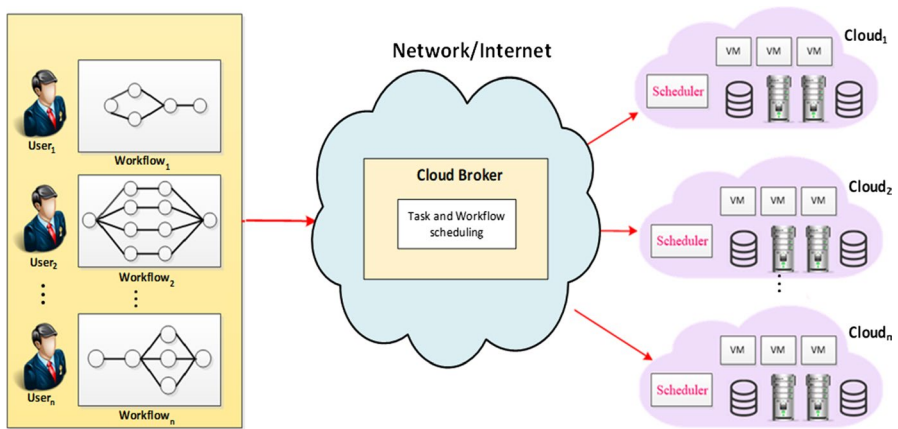
<sup>1</sup> Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran

<sup>2</sup> Afagh Higher Education Institute, Urmia, Iran

various contexts puts a heavy demand on the cloud DCs and this [2, 3] finally may lead to problems such as resource contention, service interruption, lack of interoperability, QoS degradation, and SLA violations. In addition, using a single cloud for service deployment is susceptible to various failures and even security attacks such as DDoS attacks, leading to service interruptions and low availability. To prevent the reliance on a single CSP, inter-cloud environments are provided which benefit from the several independent or cooperating clouds.

By using multiple clouds instead of one, the applications and services that need horizontal scaling can be spanned on the participant clouds in the inter-clouds with different pricing models [4, 5]. Thus, by providing users with a variety of virtual resources from different IaaS platforms, their QoS requirements can be satisfied [6, 7]. The inter-clouds may consist of organizations such as cloud federations, multi-clouds, and hybrid clouds. Task and workflow scheduling problems are known to be NP-complete problems [8]. In environments that utilize multiple independent clouds, the constituent clouds may interact or cooperate with each other in different topologies. Efficient resource management is one of the main challenges in both single-cloud and inter-cloud environments [9, 10]. Task and workflow scheduling is one of the effective tools to satisfy customers objectives in terms of QoS, cost, and performance while aiding the CSPs by reducing SLA violations (SLAV), reducing power consumption costs, providing green computing, more return on investment, and so on [11–13]. Numerous task and workflow scheduling schemes are presented in the literature, which regarding their target environment can be categorized as scheduling schemes devoted to single cloud and scheduling schemes dedicated to inter-cloud environments. Scheduling algorithms single-cloud environments try to assign user-submitted tasks to the proper VMs by considering constraints such as deadlines and cost while in the inter-cloud environment each task should be assigned to a proper VM in a proper try to map the user's submitted tasks onto various heterogeneous virtual resources dispersed in different CSPs regarding some constraints [14]. Scheduling problems are widely studied in the single-cloud environment, and various heuristic [15–25] and metaheuristic [26–31] scheduling schemes are provided in the cloud computing literature. With the increasing trend toward the inter-cloud environments, the scheduling problem is focused by research communities and numerous state-of-the-art task and workflow scheduling solutions for the inter-cloud environments are provided aiming to better benefit from computing and storage infrastructure provided by them. Figure 1 indicates a sample broker-based architecture for inter-clouds which can be considered in such environments. However, there is a lack of review article, to present a comprehensive survey and taxonomy of the inter-cloud scheduling schemes and highlight their capabilities and features.

Motivated with these issues, this article puts forward a comprehensive analysis and survey of the scheduling frameworks provided for the inter-cloud systems. To be more specific, it first presents the taxonomy of the inter-cloud environments and specifies their challenges as well as their functional and non-functional properties. It then provides general background knowledge about the task and workflow scheduling and classifies various methods and features that are considered in the scheduling process. Then, a classification of the investigated inter-cloud



**Fig. 1** A broker-based architecture for inter-clouds

scheduling schemes is introduced based on the type of inter-cloud environment which each scheme is designed for. Moreover, it provides an extensive survey of these scheduling approaches and focuses on the state-of-the-art solutions applied to effectively and efficiently assign users tasks and workflows to the various distributed virtualized resources of the participant clouds in the inter-cloud. Besides, a comparison of the evaluation metrics, simulator software, scientific workflows, their properties, and advantages are presented which can illuminate the less investigated area that can be further focused on future research directions. To the best of our knowledge, this article is the first one solely dedicated to reviewing the inter-cloud scheduling frameworks. The main contributions of this paper are as follows:

- Basic concepts and features regarding inter-cloud environments and task and workflow scheduling schemes are provided which are essential for a better understanding of the studied scheduling frameworks.
- A taxonomy of the inter-cloud task and workflow scheduling frameworks are provided according to the environment of each scheme. Then, a skeptical review of these schemes is presented and their various features are explored and discussed.
- A comprehensive comparison of the investigated schemes is provided which is very crucial for highlighting limitations of current studies in the scheduling context and finding areas that should be further focused in future researches.

The rest of this review paper is structured as follows: Sect. 2 provides the principal concepts about the inter-cloud environments and scheduling of tasks and workflows; Sect. 3 presents the taxonomy and a survey of the inter-cloud schemes. Meanwhile, Sect. 4 provides a comparison of the studied schemes, and at last, Sect. 5 puts forward the concluding issues and future research areas.

## 2 Inter-clouds environment

By definition, inter-cloud is an interconnected cloud of clouds that are unified based on open standard protocols for cloud interoperability. Figure 2 indicates a classification of the inter-cloud systems. As shown in this figure, cloud federation and multi-clouds are two main categories of inter-cloud environments. In general, a federation cloud is an inter-cloud where a set of CSPs are interconnected to enable interoperability as well as sharing and exchanging their virtual resources with each other to present a single pool of resources to cloud customers. Federation can provide various cloud services [32] and can be used for the collaboration of governmental or private clouds. Federation clouds can be further classified as follows:

- *Peer-to-peer federation* In this case, each participant cloud in the federation is a peer cloud which collaborates with others without any mediator. In this type of cloud federation, there is no central component, but distributed entities may be employed for brokering.
- *Centralized federation* In this type of cloud federation, the participant clouds in the federation benefit from a central entity that assist them to perform resource registering, sharing, and brokering [19].

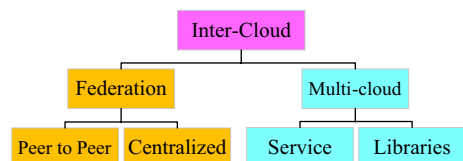
On the other hand, a multi-cloud contains multiple independent clouds that have no volunteer interconnection or sharing of their virtual resources. As a result, in such environments, managing virtual resources provisioning and scheduling is the responsibility of clients [33]. Generally, in multi-cloud environments, the portability of the applications between the clouds is of high importance. The following types of multi-cloud are considered in the literature [34, 35]:

- *Multi-cloud Services* In this case, a user may access the multi-cloud environment by using a service hosted by the cloud client externally or in-house and contain broker components.
- *Multi-Cloud Libraries* In this case, a user himself should apply its own broker via a unified cloud API as a library to access and benefit from the multi-cloud environments.

The advantages of the inter-cloud systems can be listed as follows:

- *Performance guarantee* Service performance can be maintained by using resources from other participant clouds in the federation.

**Fig. 2** Classification of the inter-clouds



- *Availability* The geographical distribution of the participant clouds in the inter-cloud allows migration of services, VMs, and data to increase the availability of services.
- *Regional workloads* Maintaining QoS is important to improve users' experience. Popular services that can be accessed from various points around the globe can direct their workloads to the regional clouds.
- *Dynamic distribution of workload* Due to geographical dispersion, it is possible to redirect workloads to clouds closer to customers. A cloud federation is an inter-cloud organization with a voluntary characteristic. It should have maximized geographical dispersion, a well-defined marketing system, and be regulated regarding the federal agreement that determines the behavior of heterogeneous and autonomous clouds.
- *Convenience* The federation provides convenience for customers about contracted services, with unified visualization of the various services available.

## 2.1 Inter-cloud challenges

These highlights focus on cloud federations and are based on motives presented in the taxonomy proposed in the work done to the comprehensive set of inter-clouds. The following main challenges can be considered for inter-cloud environments [19]:

- *Resource management* Cloud computing benefits from elasticity to adaptively provision virtual resources to deal with load fluctuations. In cloud federations, each associated CSP can offer their idle virtual resources to other CSPs in the federation as well as requesting various virtual resources from them according to cloud pre-established rules defined by a contract. Prediction accuracy of the applied methods for workload estimation is a challenging issue that CSPs should deal with it. Heterogeneity of cloud resources, the discovery of cloud resources and services as well as provisioning these resources are of the issues which should be challenged in the inter-cloud environments.
- *Economic barriers* In general, each CSP tries to increase its profits. However, because of the lack of standards, CSPs may use proprietary data storage methods, resources management protocols, and GUI. This leads to the lock-in problem in which customers will be confined to specific CSP and may face technical and monetary costs for migrating from one CSP to another.
- *Legal issues* Some customers and organizations are restricted to the use of commercial clouds. In federations, data location or destination can be defined in advance. This behavior can be used to overcome the legal constraints faced by some institutions.
- *Security* In inter-cloud environments, the customers and clouds must trust each other, but trust establishment is complex and there is a need for the trustworthiness evaluation mechanisms in the federation. Some of the existing inter-cloud schemes employ X.509 certificates for authentication purposes. In this context, having a single sign-on authentication method in the inter-cloud environments can be very useful.

- *Monitoring* In the cloud federations, clouds' resources should be monitored and their related data must be collected and analyzed to determine the need for load distribution or other management. For this purpose, a monitoring infrastructure is required to collect and process monitoring data.
- *Portability* Inter-cloud VM mobility must not prevent the control of the clouds on their resources. Also, according to data portability property, users should have control over their PaaS and SaaS applications' data to be able to move them from one cloud to another.
- *Service-Level Agreement* In the federated clouds, each participant cloud has its own SLA management method, but having global SLAs between a federation and its customers is an ideal issue in these environments.

## 2.2 Cloud federation properties

This subsection specifies the functional and non-functional properties of the cloud federation architecture, found from the related cloud literature.

### 2.2.1 Functional properties

From the cloud federation literature, the following functional properties are achieved [1]:

- *Access* In cloud federations, CSPs employ each other's virtual resources. But, to enable this, the required access credentials to the relevant users and CSPs must be issued and the required authentication and authorization mechanisms among heterogeneous CSPs should be decided.
- *Business model* To access the virtual resources of the CSPs, the required business models among CSPs should be considered.
- *Contracts* The contracts have a direct relationship with the business model, but extend their scope beyond maintenance parameters of the business model. SLAs are contracts between cloud providers and external customers that act as a guarantee of service fulfillment. In cloud federations, there is the FLA which contains recommendations, rules, and other items that determine the behavior of clouds toward the organizations.
- *Integrity* describes the consistency of the environment regarding the offer and demand of resources by cloud providers in the federation. This integrity is an important property because the federation without it can be uncharacterized, becoming just another organization of multiple clouds. For example, without this property certain cloud providers can only consume resources and not offer any.
- *Interoperability* In a cloud federation, interoperability is needed for data exchange and resource sharing among different domains using methods such as ontology, brokering, and standard interfaces.
- *Monitoring* This can be performed for the federation and its applications, which the former try to maintain the organization of federation and the latter focus on the federation applications/services running on its infrastructure.

- *Object* The object is the unit of service that a CSP can present and it will be used among the participants of the federation when resource consumption is needed.
- *Provisioning* Provisioning consists of the decision and coordination of the distribution of applications to external customers through federation providers. These actions consider the installation, planning, migration of application components when convenient or necessary, match making, data replication, etc.
- *Service manager* Service management discovers the presented services in the cloud federation.

### 2.2.2 Non-functional properties

These properties are not directly related to the services presented in the federation, but those related to the usage behavior of the environment. Non-functional properties for the cloud federation are as follows [1]:

- *Centric* It indicates the focus of implementation and usability of some elements in their architecture and makes it possible to find the features that the federation tries to increase.
- *Expansion* This property reflects the expansion of a federation horizontally, vertically or a hybrid of them.
- *Interaction architecture* The interaction of external users with the federation can be performed centrally through a single access point, or in a decentralized way using each cloud.
- *Practice niche* The niche property describes the profile of customers about the payment of consumed resources.
- *Visibility* Determines how the federation is seen by the external users. Transparent visibility does not reveal the structure of the federation, but with translucent visibility, the external user can be aware of the federation architecture.
- *Volunteer* Participation in a cloud federation should be voluntarily, and each cloud should have some knowledge about the federation structure to be able to cooperate with other clouds.

## 3 Scheduling schemes

Figure 3 depicts the various features of the task and workflow scheduling schemes. As shown in this figure, the evaluation and analysis of the task and workflow scheduling schemes can be conducted on the real systems or by using simulators software like CloudSim. Generally, the scheduling process is aimed to assign the required tasks to the appropriate VM. To handle the client's requests, scheduling schemes can benefit from homogeneous or heterogeneous VMs, in which in the second case, scheduling methods should select an appropriate type of VM for each task. Moreover, the CPU considered for each VM can be single core or multi-core. Besides, some of the scheduling schemes may also try to select the PM where the VM finally will be placed on. Moreover, based on the number of sites or DCs, applied in the cloud, scheduling approaches can be classified as a single site or multi-site scheduling schemes, in which



Fig. 3 Task and workflow scheduling features



in the latter, the scheduling approach should also select the site or cloud on which the tasks should be placed and run. The user's submitted tasks can be classified as computationally intensive or data intensive, in that in the second case, the applied data can be non-replicated, replicated, local, or remote. Data-intensive scheduling schemes deal with a large volume of the data which may be fixed located in the predefined location, or they may be movable by incurring some monetary costs and overheads. Moreover, based on the number of workflows, scheduling schemes support, they can be classified as single or multiple-workflow schemes. Furthermore, in the workflow scheduling approach, the task execution order may be specified as scheduling output. While most scheduling schemes consider static power management, some others may benefit from the DVFS-based dynamic power management which reduces the operating frequency of CPU to mitigate its power consumption while meeting the deadline. Besides, some of the scheduling solutions consider the monetary cost of the task and workflow scheduling which may rely on the fix or dynamic pricing. In addition, regarding the security support in the scheduling solution, they can be classified as insecure scheduling and secure scheduling schemes.

## 4 Proposed inter-clouds scheduling schemes

A number of task and workflow scheduling frameworks such as [31, 36–47] are designed for various inter-cloud systems. This section is aimed to illuminate the main features and capabilities of these scheduling frameworks.

- SWFs applied in the studied solutions to analyze the merits of the schemes.
- Evaluation factors employed in the simulation process.
- Simulators and CSPs applied in the evaluation phase of the task and workflow scheduling schemes

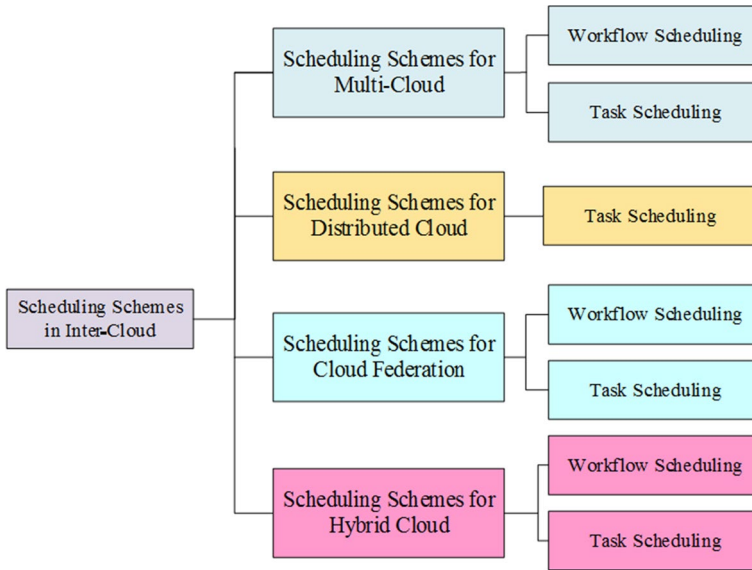
Figure 4 exhibits the classification of the scheduling schemes and the proposed solution provided for the various multi-cloud environments. As shown in this figure, we first classify the proposed scheduling approaches based on the environments designed for them. Afterward, we further classify them according to the task dependency and categorize them into task scheduling and workflow scheduling branches.

### 4.1 Multi-cloud Scheduling schemes

This subsection investigates the scheduling approaches such as [19–25, 30] designed for multi-cloud environments.

#### 4.1.1 Workflow scheduling in multi-cloud

In [48], Gupta et al. proposed a workflow scheduling approach for multi-clouds that considers the transfer time. It computes the B-level priority of the tasks and conducts the VM selection regarding the computed B-level priority of the tasks. But it



**Fig. 4** Classification of the inter-cloud scheduling schemes

does not support preemptive scheduling in a multi-cloud environment. They conducted their experiments on standard SWFs and indicated that their scheme is able to outperform the HEFT and improves makespan and resource utilization.

In [49], the authors provided a priority-based two-phase workflow scheduling algorithm for heterogeneous multi-cloud systems that can deal with large workflows. It considers factors such as computation costs and computation costs for priority and then maps the tasks according to their priority to the proper VMs. The authors analyzed their approach by performing their experiments on the standard SWFs and compared it with the other scheduling methods such as HEFT and FCFS. They indicated that their approach outperforms other algorithms based on speed-up, average cloud utilization, and makespan.

The work in [50] presented a broker-based solution for workflow scheduling in a multi-cloud environment, which enables choosing a target CSP using a service broker for and manages workflow data regarding customers' SLA requirements. However, they do not consider the data locality of the multi-cloud systems in their approach.

Lin et al. [51] presented MCPCP, a workflow scheduling approach which provides an efficient method to mitigate the workflow execution costs while meeting the specified deadlines. In this process, they have adapted PCPA or partial critical paths algorithm for the multi-cloud environment. It considers cost per time interval, different instance types from various CSPs, homogeneous intra-bandwidth, and heterogeneous inter-bandwidth. The authors indicated that workflow scheduling in the multi-cloud can outperform scheduling with a single cloud, even with low bandwidth communication links. Nevertheless, they did not explore the impact of execution time accuracy and fluctuations.

In [52], the authors introduced a multi-objective PSO-based scheduling approach for SWF in the multi-cloud environment, intending to decrease the makespan and cost while increasing reliability. The cloud computing environments may suffer from transient faults, which may result in the workflow execution failure. In this scheme, the authors considered the probability of failure by a Poisson distribution and the probability of successful task execution using an exponential distribution. This scheduling scheme considers tasks execution location and their data transmission order. This scheme first selects the IaaS platform; then, it chooses the VMs' type and determines the order of data transmission among tasks. This scheduling approach operates based on the PSO algorithm and takes into account the location of tasks execution and the order of the tasks. They considered a task queue for each CSP and assumed an infinite number of VM to be accessible for CSP users. In addition, for each CSP a pricing model and some performance metrics are applied. The authors have conducted their evaluations in Python and applied SWFs such as LIGO, Montage, CyberShake, and SIPHT in their experiments. They indicated that their algorithm is able to outperform the CMOHEFT (constraint-MOHEFT which is a modified MOHEFT algorithm by authors) and RANDOM scheduling algorithms.

In [53], the author provides a pricing model and a truthful task scheduling and applies factors like monetary completion time and cost.

Chen et al. [54] designed an adaptive task scheduling approach that considers the flexibility of precedence constraints among tasks. It deals with multiple-workflow scheduling using a rescheduling heuristic which supports task rearrangement. The scheduling of several workflows simultaneously is a challenging issue due to the heterogeneity of the distributed resources located on different sites and resource contention among them which prevents effective resource utilization. This scheme considers the uncertainty in the resource performance predictions that have a negative impact on the robustness of the scheduling method. By performing experiments, they indicated that despite some inaccuracies in the predictions their rescheduling approach can provide better results.

In [55], the authors put forward a multi-site workflow scheduling that applies performance models to forecast the execution time and conduct adaptive probes to detect the network throughput among sites. They employed applications by applying the Swift which is a script-based framework for parallel workflow execution and used two distributed multi-clusters multiple clouds. The authors indicated that their approach is able to enhance the utilization of resources and mitigate the execution time. The advantages include introducing the workflow skeletons concept and extending the SKOPE approach for modeling data movement and computations of the workflows.

The scheduling approach in [56] extends PCP and provided a communication-aware approach for data-intensive workflow scheduling for the multi-cloud environments. It modifies the partial critical paths to reduce the execution cost and meet the deadlines. They divided each workflow into some paths in which tasks in each path have more data exchange among themselves and low data exchange with other paths. It is evaluated using three synthetic SWFs.

Zhang et al. [57] provided a DC selection solution to decrease the number of DCs while having enough storage for SWF execution. They also improved inter-DC

network usage for the data transfer of SWFs and put forward a data placement algorithm by applying k-means to place the SWF data and lessen the initial data transfer among DCs. In addition, a multilevel task replication scheduling method is provided to mitigate the data transfer among DCs in the SWF executions.

Table 1 exhibits the factors and workflows applied in the simulation process and the simulation environments and software utilized in the workflow scheduling solutions designed for the multi-cloud environments.

#### 4.1.2 Task scheduling in multi-cloud

In [34], Panda et al. presented ATS, an allocation-aware task scheduling algorithm for heterogeneous multi-cloud. It consists of matching, allocating, and scheduling steps while aiming to mitigate makespan and tasks rescheduling. The authors performed their experiments using the synthetic datasets and analyzed metrics such as the average utilization of the cloud resources and makespan.

In [35], the authors provided a multi-objective approach for scheduling of tasks in the heterogeneous multi-cloud environments, which decreases the makespan and cost as well as enhancing resource utilization. Their proposed algorithm applies the CMMS or cloud min–min scheduling and PBTS or profit-based task scheduling which are two previously proposed scheduling approaches. It mainly tries to reduce the tasks of scheduling cost and makespan. They conducted simulations on the benchmark and synthetic scheduling datasets and exhibited that their algorithm is able to balance cost and makespan of schedule better than the PBTS and CMMS.

The work in [58] proposed three task scheduling algorithms, called MCC or minimum completion cloud, MEMAX or median max, and CMMN or cloud min–max normalization for heterogeneous multi-clouds to mitigate the total execution time and increase the average cloud usage. The first one of these algorithms operates in a single phase while the two others operate in two phases. The MCC algorithm finds the completion time of the existing tasks on all clouds and assigns them to the clouds that have the lowest completion time. In this scheme, MEMAX first computes the average execution time of all tasks on all clouds. In the second step, it chooses the task with a maximum average value and allocates it to the cloud which results in the lowest execution time. They conducted simulations with the MATLAB software using different benchmark and synthetic datasets to evaluate makespan and average resource utilization. They indicated that CMMN better can reduce the makespan than others and the MEMAX can lead to better resource utilization than others. Also, these algorithms can outperform the base Min–Min and Max–Min scheduling algorithms.

In [59], Tejaswi et al. provided a GA-based approach for task scheduling problem, which applies makespan as its fitness function. They assumed that expected time to run each task on each cloud and also the dependency among tasks is known. The initial population is randomly generated in which each gene in a chromosome denotes a task that is allocated to one of the clouds. The authors have evaluated their algorithm using different benchmark datasets using C++ and MATLAB software and analyzed the makespan achieved by their approach. In these simulations, they considered 512 and 1024 tasks to be scheduled on the 16 and 32 clouds.

**Table 1** Comparison of the workflow scheduling schemes provided for multi-cloud environments

Schemes	Simulator/environment										Evaluation factor					Property									
	Workflow	Mon- tage (Ligo)	Inspir alEpi g- nom- ics	Cyber- Shake	SIPHT	WIEN2 k LAB	MAT- LAB	Ama- zon EC2	Python	Grid- Sim	Micro- soft Azure	Energy	Aver- age speed- up	Makes- pan	Cloud Utili- zation	Execu- tion Time	Pro- cess- ing Time	Cost	SLR	Resource- aware	Cost- aware	Load- aware	SLA- aware	Allo- cation- aware	
[48]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[49]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[50]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[51]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[52]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[53]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[54]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[55]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
[56]	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

In [60], the authors proposed a power-aware scheduling approach for decentralized clouds which considers resource failures to prevent deadline violations. They introduced two methods for weighing conflicting objectives to consider their significance to produce a power efficiency and resource utilization. The advantages include reducing power consumption, job rejections, and deadline violations.

The approach provided in [61] presented CYCLONE, a platform for SWF in the heterogeneous multi-clouds that considers issues such as access control, data protection, and security infrastructure for distributed processing of bioinformatics data and distributed cross-organizations scientists. The CYCLONE is created based on the main features of the SlipStream cloud management platform and enables the provision of inter-cloud network infrastructure and security infrastructure.

Frincu et al. [62] provided a scheduling algorithm aimed to achieve fault tolerance, high availability, resource utilization, and mitigating cost. This scheme uses a multi-objective SA to maximize resource usage, and application's fault tolerance and availability via proper distributing its component on the allocated nodes, reducing running cost by optimizing the component to host mappings via rescheduling the component when the host load is high or low. Their algorithm is compared with the round robin scheduling algorithm on social news applications and synthetic loads and costs.

The approach provided in [63] introduced MORSA, a multi-objective real-time scheduling framework to decrease deadlines and mitigate job execution costs. The game theory is employed for the truthful information provided by the CSPs. This scheme is analyzed with synthetic and randomly produced tasks. As an advantage, the proposed approach reduces the execution times, compared with the MOEA.

In [64], Kang et al. presented a scheduling scheme to handle multiple divisible workloads on the multi-clouds, which considers scheduling factors such as the release times, ready times, computation of loads, and the topology of the network according to the dedicated links. They applied a step-based multi-round scheduling approach in the static and dynamic scheduling approaches that the first one assumes the release times of VMs are known, while the other considers that it is unknown until they are released. Although a prediction model can be used to predict the release time of computing nodes, it is not applied in this scheme.

In [65], Panda et al. presented UQMM, an uncertainty-based QoS-aware Min–Min algorithm that considers uncertainty-based QoS factors in a heterogeneous multi-cloud environment. This algorithm is evaluated on the synthetic workflows and also two SWFs. The authors compared their approach with the min–min, min–max normalization, and smoothing-based scheduling methods and evaluated its results by using statistical tests such as ANOVA and *t* test. But, this algorithm does not consider execution modes, such as best effort and advanced reservation.

The scheduling approach in [66] provides three task scheduling algorithms, denoted as AMinB, AMaxB, and AMinMaxB, for heterogeneous multi-cloud systems which are extensions of the Max–Min and Min–Min algorithms. These algorithms perform matching, allocating, and scheduling steps. The allocating step fills the gaps between matching and scheduling. Multi-cloud collaboration makes mapping more efficient than others. The authors conduct their experiments on two benchmark and synthetic datasets and realize that their scheme is able

to reduce makespan and resource utilization and throughput in comparison with the Max–Min, Min–Max, and Min–Min methods. But, they did not analyze the execution cost, transfer cost, and the deadline of their approach.

In [67], the authors propose three task partitioning scheduling algorithms for the heterogeneous multi-cloud environment, denoted as CTPS, CMMTPS, and CMAXMTPS, in which the first one is an online algorithm and two others are offline. It comprises two preprocessing and processing steps for scheduling a task in two clouds aiming to reduce the scheduling makespan and increase utilization of resources. The preprocessing completion time of the tasks is calculated over all the clouds. Then, a task is selected and assigned to a cloud. They have considered factors such as resource utilization and makespan in their experiments. But, they have not considered the communication time between the preprocessing and processing phases and the cost of the transfer and execution time.

The introduced approach in [68] presented GACCRATS, GA-based user-aware task scheduling and resource allocation for multi-cloud systems. TLBO is applied to extract the task–VM pair having minimum makespan time, whereas GACCRA TS tries to minimize the makespan of tasks and maximization of user satisfaction rate. The algorithm performs mapping and scheduling the tasks. It operates in two steps, in which in the first step, it performs GA-based resource management, and then, it conducts the shortest job first scheduling.

In [69], the authors put forward DSS, an adaptive scheduling method for multi-cloud systems that combines the divisible load theory and node availability forecasting methods to provide high performance. The advantages include using a forecasting method for estimating the ready time of VMs and processing time of the workloads using previous data about processing times. For this purpose, they assumed that a dataset is located on each VM for training in which its size will increase as more processing loads enter the system. Their proposed architecture for the multi-clouds supports several geo-scattered gateways, in which each of them has a scheduler that is responsible for receiving loads and scheduling them on the VMs. The gateways share a pool of VMs that have various computational and communication features. Also, the gateways may be connected to all nodes using links with various capacities. They indicated that their scheduling approach is able to reduce the tasks' processing time.

In [70], Heilig et al. proposed BRKGA-MC, a biased random key GA for resource management in the multi-clouds which finds a proper configuration for resource, regarding the different IaaS offerings to satisfy user demands. This algorithm determines a feasible approach with excellent quality making it suitable for being included as a decision support tool in cloud brokerage. It decreases costs and execution time. The costs consist of the VM purchasing costs and the communication cost between tasks located in various VMs. Nonetheless, the location of CSPs is not considered in this solution.

Table 2 shows the evaluation factors applied in the simulation process and the simulation environments and software utilized in the task scheduling approaches designed for the multi-cloud environments.

**Table 2** Comparison of the task scheduling approaches provided for the multi-cloud environments

Schemes	Simulator/environment				Evaluation factor						Property										
	MAT-LAB	Ama-zon EC2	Python	Grid-Sim	C++	Cloud-Sim	CPU Usage	Num-ber of Task	Makes-pan	Cloud Utiliza-tion	Cus-tomer Satis-fac-tion	Pro-cess-ing Time	Cost	Rejected Applica-tions	Cost- aware	Allo- cation- aware	Resource- aware	Energy- aware	Data- aware	SLA- aware	
[35]	✓							✓				✓									
[34]	✓							✓					✓								
[58]	✓							✓							✓						
[59]	✓				✓			✓						✓							
[60]														✓							
[61]																					✓
[62]		✓											✓								
[63]											✓				✓						
[64]												✓					✓				
[65]	✓							✓													✓
[66]	✓							✓													✓
[67]	✓							✓													✓
[68]	✓							✓													✓
[69]		✓																			✓
[70]		✓																			✓



## 4.2 Scheduling schemes for distributed clouds

This section is aimed to provide a discussion on the scheduling solutions such as [33, 71, 72] provided for distributed cloud in the scheduling literature.

### 4.2.1 Task scheduling in distributed clouds

In [73], Alsughayyir et al. presented EAGS, a workflow scheduling approach to decrease the power usage in a decentralized multi-cloud which contains geo-distributed heterogeneous clouds which are provided by various CSPs.

When this scheme receives an application, it computes critical path execution time using the best VM which can be offered to the user application. When this execution time is less than the specified deadline, the application will be accepted for scheduling. In the scheduling process, the tasks can be assigned to the local resources or remote resources in other clouds. To mitigate power consumption, this workflow scheduling framework benefits from DVFS. The authors have evaluated their approach with the min–min scheduling algorithm and indicated that their approach can reduce power users better than the Min–Min method.

Kessaci et al. [74] presented MO-GA, a multi-objective GA with a Pareto approach to mitigate the power usage, CO<sub>2</sub> emissions in geographically distributed CSPs. They put forward a greedy heuristic approach to increase the number of scheduled jobs. This scheme is analyzed using load traces of various instances from Feitelson's parallel workload archive. The advantages include mitigating the power usage by applying DVS within the DC and enabling delays for the applications by using a pricing model with penalties.

The work in [75] introduced near-optimum scheduling considering the heterogeneity of multiple DCs in a CSP which takes into account factors like power cost, CO<sub>2</sub> emissions, load, and CPU energy efficiency. This scheme can decrease power usage to increase profit and carbon emissions by applying managing CPU voltage via DVS or dynamic voltage scaling. However, they have not considered the energy and delay of moving datasets between the DCs in data-intensive applications. In addition, they have not analyzed the profit and power sustainability in the VM allocation.

The scheduling framework in [76] put forward an approach for multi-resource allocations and multi-job scheduling in the distributed and heterogeneous clouds. This method mitigates the makespan and increases resource utilization, and achieves a joint solution for the distributed cloud job scheduling. This scheme consists of three components: job scheduling component, resource allocation component, and infrastructure management component. Nevertheless, they did not consider load balancing in the distributed clouds.

Yin et al. [77] presented a two-step scheduling algorithm for geographically distributed DCs in which in the first step, it groups tasks regarding the data sharing relations among them and models this relation using hypergraphs, partition tasks into groups by using the hypergraph partitioning and dispatches the achieved groups to the DC to benefit from data locality. Then, it trades off the completion time in DC based on the relations between tasks and groups. In the second step, it balances the completion time across DCs regarding relations among groups and tasks. They used

the China-Astronomy-Cloud model for their experiments and indicated that their scheduling approach can reduce the makespan and the transferred data. However, they have not considered more complex, heterogeneity, and realistic conditions in the network.

The introduced approach in [78] presented AMP, an ACO-based approach for the geographically distributed DCs, which increases the profit and considers the users' and CSPs' satisfaction. AMP formulates the maximization of resource utilization problem as an optimization problem regarding DC's and customers' constraints with binary integer programming. The authors designed a low complexity, requests dispatching, and resource management approach to solve the large-scale problems using the ACO algorithm and considered the capacity constraints of DCs. The advantages of the AMP include increasing CSP profit via dispatching service requests to the appropriate DCs, and Table 3 specifies the evaluation factors applied in the simulation process as well as the simulation environments and software utilized in the task scheduling schemes designed for the distributed cloud environments.

### 4.3 Scheduling schemes for cloud federation

This section is intended to provide a discussion on the scheduling solutions such as [79, 80] provided for cloud federation in the scheduling literature.

#### 4.3.1 Workflow scheduling in cloud federation

In [81], the authors investigated the scheduling of the large workflow on federated clouds based on the QoS factors. They have used SMARTFED for the cloud federation and applied their algorithm to schedule various workflows based on the QoS parameters such as resource utilization, cost, makespan on the cloud federation. They have applied SmartFed which is a cloud federation simulator, built on the CloudSim. It adds the required packages for federation DC, allocator, Queue management, and storage while importing packages for DC, cloudlets, and broker from CloudSim. For VMs allocation in DC, they used a round robin algorithm. In addition, for workflow scheduling in federated clouds the WorkflowSim simulator is applied which is open source and can model workflows in the form of XML files as DAX.

Coutinho et al. [82] provided GraspCC-fed, an approach to produce the optimal estimation of the required number of VMs for each workflow in the single CSP and federated clouds. GraspCC-fed is based on the Greedy Randomized Adaptive Search Procedure (GRASP) that predicts costs according to the workflow execution time and cost. GraspCC-fed specifies the required configurations for workflow execution and the number of VMs for each CSP that must be allocated for the execution of workflow regarding users' constraints. They extended their scheme to handle workflow executions in the federated clouds and combined it with the SciCumulus workflow engine for this purpose.

In [83], the authors proposed MOHEFT, a multi-objective list-based scheduling algorithm for SWFs. It extends the HEFT workflow scheduling method to achieve

**Table 3** Comparison of the task scheduling schemes provided for distributed clouds

Schemes	Simulator/environment				Evaluation factor							Property						
	MAT-LAB	Ama-zon EC2	Python	Grid-Sim	Cloud-Sim	Mem-ory Usage	Num-ber of jobs	CPU Usage	Makes-pan	Load	Profit	Exe-cution Time	Cost	Energy Con-sump-tion	Energy-aware	Cost-aware	Resource-aware	SLA-aware
[73]					✓				✓				✓	✓				
[74]	✓									✓			✓	✓		✓		
[75]	✓								✓					✓			✓	
[76]								✓									✓	
[77]					✓							✓					✓	
[78]	✓									✓								✓

near-optimum Pareto front via the crowding distance, to reduce cost and makespan. They evaluated MOHEFT with real-world and synthetic workflows using the HV metric and compared it against the HEFT and SPEA2. The authors indicated that MOHEFT obtains better makespan and cost. To be more specific, when a workflow has a few parallel tasks, MOHEFT and SPEA2 provide better results than HEFT, while for balanced workflows with more parallel tasks, MOHEFT outperforms SPEA2. They conducted their experiments using GoGrid and Amazon EC2 and indicated that when data communications are more than the computation time, the cloud federation cannot be well benefited.

Table 4 specifies the evaluation factors and workflows considered in the simulation process of the workflow scheduling schemes designed for the cloud federation environments. In addition, the employed simulation environments and software utilized in these scheduling approaches are highlighted.

### 4.3.2 Task scheduling in cloud federation

Various task scheduling solutions such as [47] are designed for federated clouds which this section is dealing with.

In [84], Nguyen et al. provided DrbCF, a decentralized broker to differentiate ratio-based job scheduling to handle the workload. The variation in cloud infrastructure is recognized comparing with the fixed equal ratio among cloud sites in a cloud federation. When the cloud capacity fluctuates, the differentiated ratio enables the cloud sites to find a proper partner running its task. Advantages include enabling the load transfer between the CSPs, enabling a limited capacity site to access other CSPs at peak demand and preventing overloading caused via external loads.

In [85], the authors introduced GWpilot, to provide decentralized, multi-user, middleware independence, adaptive brokering, compatible with legacy programs, and the efficient execution of tasks. This solution can instantiate VMs depending on the available resources, allowing users to consolidate their resource provisioning. However, regarding the cost models applied in this scheme, it does not support the public CSPs.

The work in [63] provided a distributed job scheduling method using game theory to handle resource management in the cloud federation. Each job contains some tasks and communication among them. This scheme groups tasks based on their communication pattern to mitigate communication latency and messaging overheads among participant clouds in a cloud federation. It also finds the Nash equilibrium to enhance the cloud federation's advantage by using a pricing strategy that considers the competition degree in resource conflicts and cost. They evaluated their approach using Taiwan UniCloud and demonstrated that the CSPs can achieve more profit by outsourcing their resources and the federated clouds can have enough resources.

In [86], Gouasmi et al. introduced an exact scheduling algorithm for cloud federation environments, to process MapReduce applications in geographically distributed clusters, and solved it as a mixed-integer program. It presents a baseline for this exact scheduling model and formulates it as a mixed-integer program. The advantages include considering data locality, cost of the VMs, and data transfer cost while meeting the deadline.

**Table 4** Comparison of the workflow scheduling schemes provided for cloud federation

Schemes	Workflow		Simulator/environment		Evaluation factor					Property							
	Mon- tage	Inspiral (Ligo) enomi- cs	Cloud- Sim	Ama- zon EC2	Grid- Sim	Num- ber of Task	CPU Usage	Makes- pan	Cloud Utili- zation	Cloud Activi- ties	Execu- tion Time	Cost	Energy Con- sump- tion	Resource- aware	SLA- aware	Cost- aware	
[82]				✓					✓				✓				
[81]	✓		✓													✓	
[83]				✓				✓							✓		✓

The approach in [87] puts forward FDMR, a geographically distributed algorithm to run MapReduce jobs in the federated clouds and mitigate the execution cost while considering deadline constraints. This work introduces an exact scheduling method using a linear integer program as a baseline to evaluate and discuss the heuristic algorithm. The reduction in idle VMs in the federation has a direct effect on decreasing the costs and job response time. In addition, the FDMR can enhance the utilization of federation resources and considers factors such as data transfer, data locality, and availability of resources. Nevertheless, their scheme does not take into account the budget constraint. The performance evaluation proves that FDMR is able to reduce job execution costs while considering deadline constraints.

Gouasmi et al. [88] presented FedSCD, a distributed scheduling scheme to handle MapReduce applications in the federation's distributed clusters. It takes into account the data locality and tries to reduce VMs and data transfer costs among clusters while regarding the specified deadlines. It decreases the cost through outsourcing the MAP tasks and splitting their related data to the idle VMs into other clusters existed in the federation. The advantages include enhancing performance considering the cost, resource utilization, and response time. This scheme performance of this scheme is evaluated against partially distributed scheduling method in a federated cloud and provides better resource management and scheduling response time while reducing job execution cost by decreasing idle VMs and preventing resource wastage. Table 5 specifies the evaluation factors applied in the simulation process as well as the simulation environments and software utilized in the task scheduling schemes designed for the cloud federation environments.

#### 4.4 Scheduling schemes for hybrid clouds

A hybrid cloud consists of a set of private clouds and public clouds. The users submit their tasks/workflows to their private clouds which belong to their organizations. When private clouds lack the required resources to execute the customers' tasks, the private cloud may decide to forward these tasks to the public clouds. In this process, one of the main objectives of the hybrid cloud scheduling schemes is to minimize the number of tasks that should be sent to public clouds for execution.

##### 4.4.1 Workflow scheduling in hybrid cloud

This subsection investigates the workflow scheduling schemes such as [11, 89–107] provided for the hybrid cloud. For instance, in [108], the authors proposed HCOC, a cost-aware scheduling algorithm for hybrid cloud to speed up the workflow execution regarding the deadlines. It supports multiple cores and can reduce the makespan. It enables the users to control costs by tuning the execution time of the workflow. If the available bandwidth between two public CSPs can be predicted, this scheme provides better results, but they have not considered this issue.

The approach in [109] presented a service workflows scheduling solution in a hybrid cloud to determine which services should use paid virtual resources and which types of the resource must be requested to reduce costs in specified deadlines.

**Table 5** Comparison of the task scheduling schemes provided for cloud federation

Schemes	Simulator/environment			Evaluation factor					Property												
	MAT-LAB	Open-Stack	FLUKA	CPLEX Solver	Java	Dif-feren-tiated rate	Execu-tion Time	Suc-cess job rate	Cloud Utiliza-tion	Profit	VM creation attempt	Cost	VM Utiliza-tion	SLA-aware	Allo-cation-aware	Data-aware	Energy-aware	Resource-aware	Data-aware	Cost-aware	
[84]	✓					✓		✓						✓	✓						
[85]			✓							✓					✓						
[63]							✓		✓		✓					✓					
[86]		✓									✓	✓					✓				
[87]		✓									✓	✓	✓							✓	
[88]		✓									✓	✓	✓							✓	✓

They have deployed a hybrid cloud that offers support for automatic service installation in the resources dynamically provided by the grid or the cloud. As an advantage, it is able to schedule workflows in a real CSP for reducing their execution costs and meeting the execution deadlines.

The work in [9] analyzed scheduling algorithms regarding the uncertainty of the communication links availability. In clouds, such precise data are hard to obtain due to concurrency in communication links, which enlarges the makespan and costs, and lack of knowledge on the infrastructure such as network topology in public clouds.

Lin et al. [110] provided HIAP, an online scheduling method for SWF on hybrid clouds which considers factors such as deadlines and cost in the scheduling process. It partitions the application into a set of dependent tasks and considers the constraints such as bandwidth, data transfer cost, and computational cost. They also considered the execution time accuracy and tried to handle the fluctuant instances.

Table 6 specifies the applied workflow, evaluation factors, and simulators applied in the simulation of the workflow scheduling schemes designed for hybrid cloud environments.

#### 4.4.2 Task scheduling in hybrid clouds

This subsection investigates the workflow scheduling schemes such as [17, 111–137] provided for the hybrid clouds. For example, in [138], Calheiros et al. designed an architecture for the cost-aware dynamic provisioning and scheduling to complete tasks' execution within their deadlines regarding the task level load. As an advantage, they also used an accounting approach to determine the share of the cost of utilization of public CSP resources to be assigned to each user.

Yuan et al. [139] introduced PMA, a profit-aware algorithm to find the temporal price fluctuations in hybrid clouds. It can schedule arriving tasks to run in the private and public clouds by combining the SA and PSO algorithms. The advantages include increasing profit and throughput of the private CSP as well as meeting delay constraints.

In [140], the authors presented TTSA, a task scheduling solution that models cost reduction as a mixed-integer linear problem. The near-optimal scheduling produced by the TTSA can increase throughput and reduce the cost of private cloud while meeting the deadlines of all the tasks. But, factors such as dispatching time and execution delay are not considered in this scheduling model.

Table 7 exhibits the simulators, environments, and evaluation factors applied in the evaluations of the task scheduling schemes proposed for the hybrid clouds.

## 5 Discussion

This subsection puts forward a comparison of the scheduling schemes in a multi-cloud environment, outlined and studied in the previous section. The result of this section can illuminate the directions of future research and help to develop new scheduling solutions for multi-cloud systems. This section provides the following information about multi-cloud scheduling schemes:



**Table 6** Comparison of the workflow scheduling schemes provided for hybrid clouds

Schemes	Simulator/environment										Evaluation factor					Property							
	Mon- taga (Ligo)	Inspir al	AIRSN	Cyber- Shake	CSTEM	SIPHT	MDP	Broad- band	Chi- mera	Epig- enomi- cs	Ama- zon EC2	Java Cloud Sim	Cost	Makes- pan	Dead- line	Execu- tion Time	Accept- ance ratio	VM Utili- zation	Cost- aware	Energy- aware	Dead- line- aware	Data- aware	
[92]													✓			✓			✓				
[98]											✓		✓									✓	
[99]										✓			✓									✓	
[100]	✓		✓						✓				✓					✓				✓	
[101]	✓		✓						✓				✓									✓	
[102]																							
[103]	✓									✓			✓				✓						✓
[105]													✓						✓				
[106]	✓									✓			✓										✓
[107]																							✓
[108]	✓									✓			✓										✓
[109]													✓										✓
[110]	✓									✓			✓										✓
[9]	✓									✓			✓										✓

**Table 7** Comparison of the task scheduling solutions provided for the hybrid clouds

Schemes	Simulator/environment		Evaluation factor					Property									
	MAT-LAB	Ama-Cloud-Sim zon EC2	Java Cloud- Sim	Makespan	Cost	Num- ber of Task	Dead- line	VM Utili- zation	Profit	Time con- ver- gence itera- tions	Execu- tion suc- cess ratio	Mem- ory Utili- zation	Dead- line- aware	Resource- aware	Cost- aware	SLA- aware	Data- aware
[130]			✓	✓	✓	✓	✓		✓			✓		✓			
[131]			✓	✓	✓	✓	✓							✓			
[132]		✓		✓	✓	✓								✓			
[133]																✓	
[134]			✓	✓	✓	✓	✓		✓								
[126]	✓		✓		✓	✓	✓					✓					
[135]		✓			✓	✓	✓	✓				✓					
[136]		✓		✓	✓	✓	✓						✓				
[120]	✓								✓								✓
[137]		✓							✓								
[126]			✓		✓	✓	✓			✓			✓				
[128]					✓	✓	✓							✓			
[138]			✓			✓	✓	✓				✓		✓			✓
[139]						✓	✓							✓			
[140]	✓					✓	✓										
[129]			✓		✓	✓	✓									✓	

- Publication year of the investigated scheduling frameworks inter-cloud.
- Evaluation factors considered in the simulation process of DC scheduling solutions.
- Environments and simulations applied to evaluate and analyze DC schemes.
- Metaheuristic and heuristic methods adapted to handle the scheduling process.
- Datasets used to evaluate the designed scheduling frameworks and algorithms.

Figure 5 exhibits the number of task/workflow scheduling schemes each year. As shown in this figure, the number of task/workflow scheduling schemes is increasing and this context can be considered as an active research area. However, workflow scheduling investigated less than task scheduling in inter-cloud environments. Thus, the workflow scheduling problem can be analyzed further in the following studies.

Furthermore, Fig. 6 depicts the number of scheduling frameworks published for various inter-cloud environments, in which as shown in this figure fewer workflow scheduling researches have been conducted in the various inter-cloud environments. Thus, further research can be conducted to more effectively evaluate workflow scheduling in different environments. Also, as shown in this figure, most of the investigated scheduling schemes are designed for hybrid and multi-cloud environments.

Figure 7 exhibits the number of schemes which have applied each kind of simulators/environment in the inter-cloud scheduling approaches. It indicates that MATLAB, Amazon EC2, and CloudSim are applied by most of the studied scheduling approaches. In addition, as shown in this figure, only a few schemes have implemented their suggested approach on the real cloud computing platforms such as OpenStack and Microsoft Azure. Consequently, in the next studies, to better verify the proposed scheduling frameworks it can be evaluated in such real environments.

Also, Fig. 8 lists the evaluation factors considered in the simulation process of the inter-cloud scheduling frameworks. As shown in this figure, factors such as makespan, cost, and cloud utilization are applied by most schemes to evaluate the outlined inter-cloud task and workflow scheduling solutions.

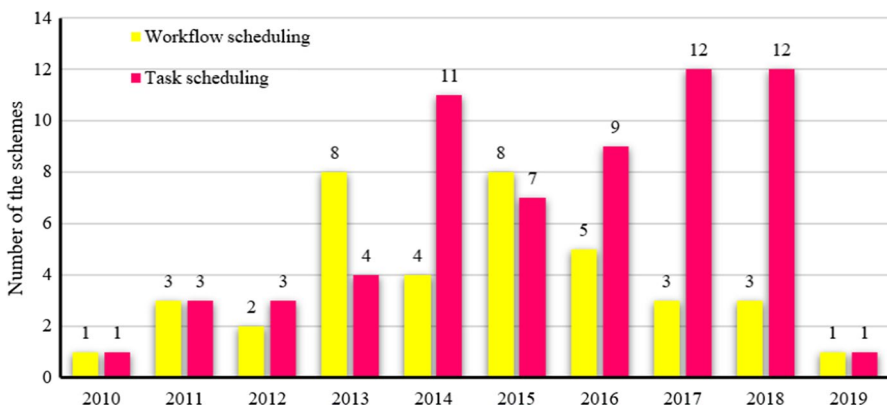


Fig. 5 Number of the workflow and task scheduling schemes in each year

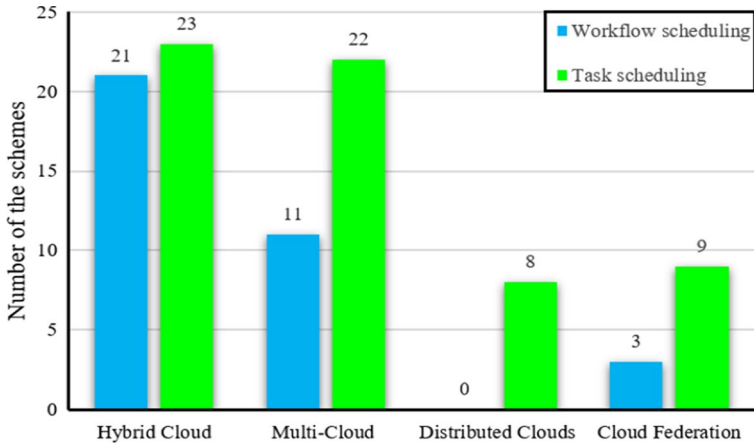


Fig. 6 Number of scheduling solutions proposed for each inter-cloud environment

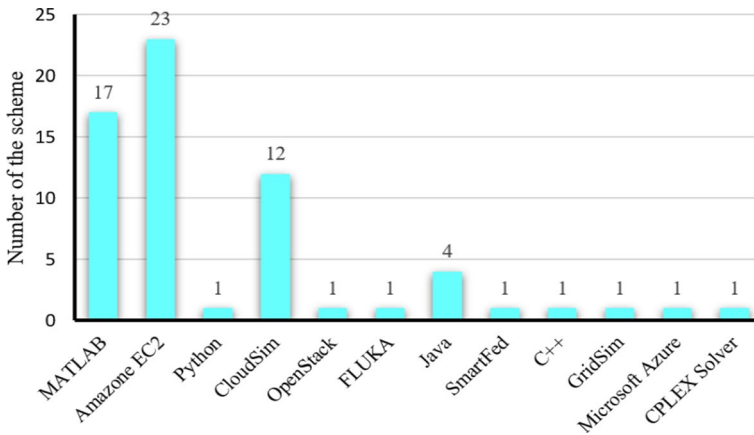


Fig. 7 Applied simulators and environments in the inter-cloud scheduling approaches

Figure 9 depicts the SWFs used in the inter-cloud scheduling schemes and the number of schemes which have applied each of them. As exhibited in this figure, SWFs such as Montage, Ligo, and Cybershake are mostly analyzed by the outlined workflow scheduling schemes provided for various inter-cloud environments. The popular scientific workflows can be described as follows:

- *Montage* It is created by the NASA/IPAC as a toolkit that can be used to produce custom mosaics of the sky using input images.
- *Cybershake* It is applied by the earthquake center of southern California to detect earthquakes in a region.
- *Epigenomics* This workflow is a data processing pipeline that contains various genome sequencing operations.

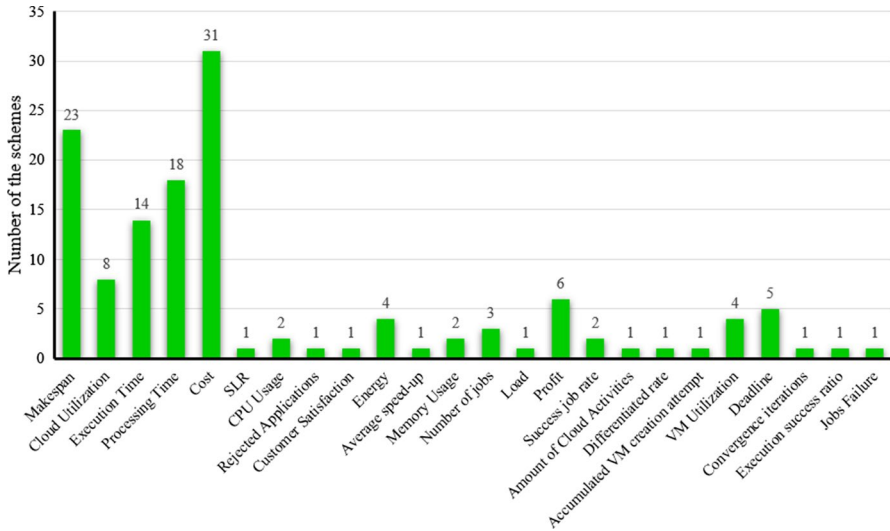


Fig. 8 Evaluation factors applied in the inter-cloud scheduling schemes

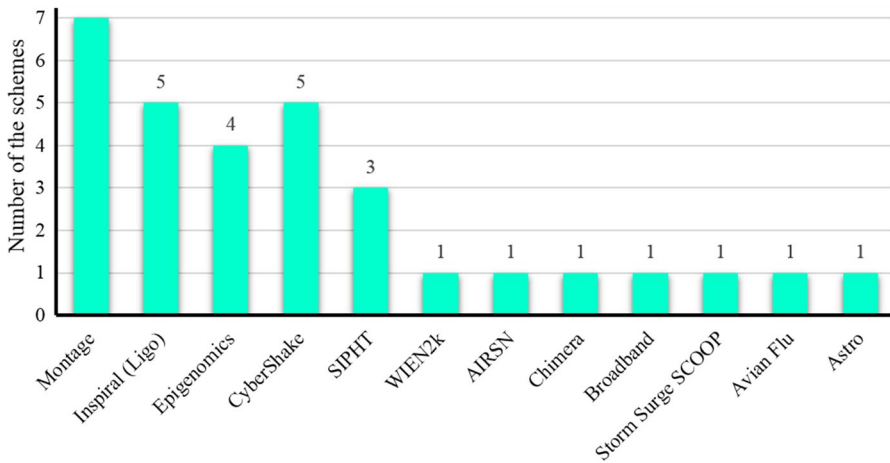


Fig. 9 SWFs used in the inter-cloud scheduling schemes

- *LIGO* This workflow is used in astronomy applications such as black holes and neutron stars.
- *SIPHT* It is used in a bioinformatics project at Harvard University.
- *WIEN2 k* It is a computationally intensive application in the material science field for performing electronic structure computation of solids. It has a balanced structure.

Figure 10 depicts the various factors which have been used in the outlined scheduling schemes and the number of schemes which have applied each factor into account in assigning tasks to the inter-cloud environments. As shown in this figure, factors such as cost, SLA, resource, and data are mostly considered by the investigated schemes. To this end, in future studies, other factors such as security and fault tolerance which are of high importance in the distributed environments should be taken into account, considering the growing of DDoS security attacks in number and complexity.

## 6 Conclusion

Effective scheduling of tasks and workflows are of high importance in various cloud computing environments, and especially in inter-cloud, which consists of numerous virtual resources in multiple clouds. Scheduling is able to reduce energy consumption by effective resource management while meeting deadlines and preventing SLAVs.

This paper focuses on the scheduling schemes provided to deal with various challenging issues in the inter-cloud environment. It first presents the background knowledge and basic concerns regarding the multi-cloud environment and scheduling of the tasks and workflows. Then, it classifies the investigated scheduling approaches based on the environments which are designed for. Also, each category is further categorized into task and workflow scheduling schemes. Besides, a skeptical review of the inter-cloud scheduling schemes is provided and their main pros and cons are

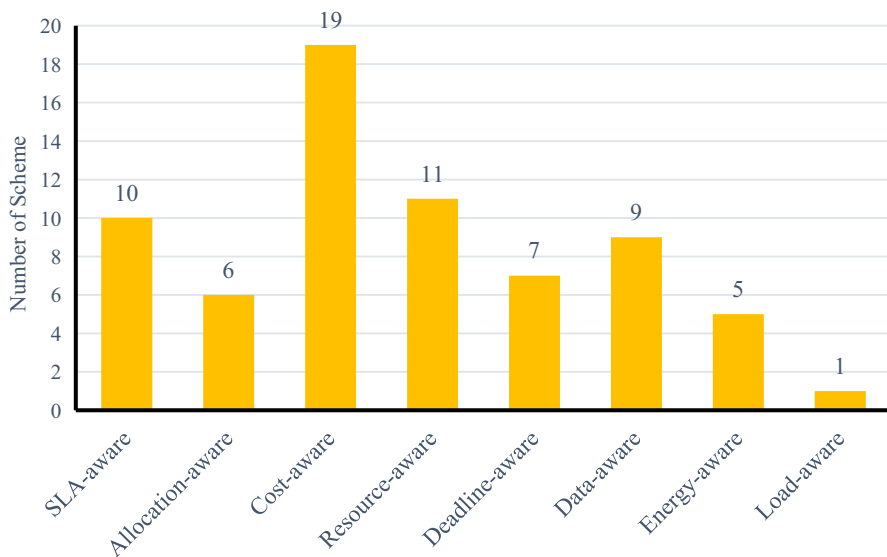


Fig. 10 Properties of the inter-cloud scheduling schemes

illuminated. Then, these schemes are compared to highlight their applied datasets, evaluation factors, and simulators.

Regarding heavy reliance on the inter-cloud schemes on the underlying network infrastructure, the reliability of data links should be investigated from two distinct aspects: first, the data links may go down; second, their bandwidth may fluctuate. Both of these problems can be handled approximately with a prediction model for the applied data links in the inter-cloud environment. Moreover, to provide reliable scheduling in inter-cloud, the failure of VMs and their performance fluctuations should be taken into account. Again, by using prediction models, the performance variations in VMs can be considered in the real-time and deadline-sensitive scheduling problems. Security is another important open issue in the distributed and network-based solutions which must be considered in the scheduling of applications in the inter-clouds. For example, in heterogeneous environments, VMs with different security levels and prices can be considered. Also, several security levels can be considered for the tasks and their accessed data in repositories. Moreover, the existence of secure links with various options can be assumed. Furthermore, each cloud in the inter-cloud environment may benefit from different security policies which can be considered in the scheduling process. For example, some clouds may prohibit to access remote data repositories or prevent the applications and workflows from transferring their data to other clouds. Regarding the before mentioned issues, conducting secure scheduling in inter-cloud environments can be further investigated in the future. Also, problems such as selfish clouds which only incur load to the inter-cloud and do not cooperate with other clouds, and various kinds of DDoS attacks should be handled in the future. Load balancing is one of the conflicting objectives which should be considered in future studies and multi-objective scheduling approaches to trade-off customer's objectives and cloud objectives, and inter-cloud objectives should be designed.

## References

1. Assis MR, Bittencourt LF (2016) A survey on cloud federation architectures: identifying functional and non-functional properties. *J Netw Comput Appl* 72:51–71
2. Masdari M, Jalali M (2016) A survey and taxonomy of DoS attacks in cloud computing. *Secur Commun Netw* 9:3724–3751
3. Masdari M, Salehi F, Jalali M, Bidaki M (2017) A survey of PSO-based scheduling algorithms in cloud computing. *J Netw Syst Manage* 25:122–158
4. Masdari M, ValiKardan S, Shahi Z, Azar SI (2016) Towards workflow scheduling in cloud computing: a comprehensive analysis. *J Netw Comput Appl* 66:64–82
5. Masdari M, Nabavi SS, Ahmadi V (2016) An overview of virtual machine placement schemes in cloud computing. *J Netw Comput Appl* 66:106–127
6. Villegas D, Bobroff N, Rodero I, Delgado J, Liu Y, Devarakonda A, Fong L, Sadjadi SM, Parashar M (2012) Cloud federation in a layered service model. *J Comput Syst Sci* 78:1330–1344
7. Giacobbe M, Celesti A, Fazio M, Villari M, Puliafito A (2015) Towards energy management in cloud federation: a survey in the perspective of future sustainable and cost-saving strategies. *Comput Netw* 91:438–452
8. Rodriguez MA, Buyya R (2017) A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments. In: *Concurrency and Computation: Practice and Experience*, vol 29

9. Bittencourt LF, Madeira ER, da Fonseca NL (2012) Impact of communication uncertainties on workflow scheduling in hybrid clouds. In: Global Communications Conference (GLOBECOM), 2012 IEEE, pp 1623–1628
10. Wu CQ, Cao H (2016) Optimizing the performance of big data workflows in multi-cloud environments under budget constraint. In: 2016 IEEE International Conference on Services Computing (SCC), pp 138–145
11. Duan R, Prodan R, Li X (2014) Multi-objective game theoretic scheduling of bag-of-tasks workflows on hybrid clouds. *IEEE Trans Cloud Comput* 2:29–42
12. Zhang M, Yang Y, Mi Z, Xiong Z (2015) An improved genetic-based approach to task scheduling in Inter-cloud environment. In: Ubiquitous Intelligence and Computing and 2015 IEEE 12th International Conference on Autonomic and Trusted Computing and 2015 IEEE 15th International Conference on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom) 2015, pp 997–1003
13. Zhang F, Cao J, Li K, Khan SU, Hwang K (2014) Multi-objective scheduling of many tasks in cloud platforms. *Future Gener Comput Syst* 37:309–320
14. Smanchat S, Viriyapant K (2015) Taxonomies of workflow scheduling problem and techniques in the cloud. *Future Gener Comput Syst* 52:1–12
15. Antony C, Chandrasekar C (2016) Performance study of parallel job scheduling in multiple cloud centers. In: IEEE International Conference on Advances in Computer Applications (ICACA), 2016, pp 298–303
16. Wen Z, Cala J, Watson P (2014) A scalable method for partitioning workflows with security requirements over federated clouds. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), 2014, pp 122–129
17. Genez TA, Bittencourt L, Fonseca N, Madeira E (2015) Estimation of the available bandwidth in inter-cloud links for task scheduling in hybrid clouds. *IEEE Trans Cloud Comput* 7:62–74
18. Lin X, Wu CQ (2013) On scientific workflow scheduling in clouds under budget constraint. In: 42nd International Conference on Parallel Processing (ICPP), 2013, pp 90–99
19. Toosi AN, Calheiros RN, Buyya R (2014) Interconnected cloud computing environments: Challenges, taxonomy, and survey. *ACM Comput Surv* 47:7
20. Cohen WE, Mahafzah BA (1998) Statistical analysis of message passing programs to guide computer design. In: Proceedings of the Thirty-First Hawaii International Conference on System Sciences, pp 544–553
21. Qasem MH, Sarhan AA, Qaddoura R, Mahafzah BA (2017) Matrix multiplication of big data using mapreduce: a review. In: 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS), pp 1–6
22. Mahafzah BA, Jaradat BA (2008) The load balancing problem in OTIS-Hypercube interconnection networks. *J Supercomput* 46:276–297
23. Mahafzah BA, Jaradat BA (2010) The hybrid dynamic parallel scheduling algorithm for load balancing on chained-cubic tree interconnection networks. *J Supercomput* 52:224–252
24. Mahafzah BA (2011) Parallel multithreaded IDA\* heuristic search: algorithm design and performance evaluation. *Int J Parallel Emergent Distrib Syst* 26:61–82
25. Al-Adwan A, Sharieh A, Mahafzah BA (2019) Parallel heuristic local search algorithm on OTIS hyper hexa-cell and OTIS mesh of trees optoelectronic architectures. *Appl Intell* 49:661–688
26. Wu C-M, Chang R-S, Chan H-Y (2014) A green energy-efficient scheduling algorithm using the DVFS technique for cloud datacenters. *Future Gener Comput Syst* 37:141–147
27. Yang J, Jiang B, Lv Z, Choo K-KR (2017) A task scheduling algorithm considering game theory designed for energy management in cloud computing. *Future Gener Comput Syst*. <https://doi.org/10.1016/j.future.2017.03.024>
28. Chang B-J, Lee Y-W, Liang Y-H (2018) Reward-based Markov chain analysis adaptive global resource management for inter-cloud computing. *Future Gener Comput Syst* 79:588–603
29. Szabo C, Sheng QZ, Kroeger T, Zhang Y, Yu J (2014) Science in the cloud: Allocation and execution of data-intensive scientific workflows. *J Grid Comput* 12:245–264
30. Masadeh R, Sharieh A, Mahafzah B (2019) Humpback whale optimization algorithm based on vocal behavior for task scheduling in cloud computing. *Int J Adv Sci Technol* 13:121–140
31. Alshraideh M, Mahafzah BA, Al-Sharaeh S (2011) A multiple-population genetic algorithm for branch coverage test data generation. *Software Qual J* 19:489–513
32. Grozev N, Buyya R (2014) Inter-Cloud architectures and application brokering: taxonomy and survey. *Softw Pract Exp* 44:369–390



33. Chen W, Xie G, Li R, Bai Y, Fan C, Li K (2017) Efficient task scheduling for budget constrained parallel applications on heterogeneous cloud computing systems. *Future Gener Comput Syst* 74:1–11
34. Panda SK, Gupta I, Jana PK (2015) Allocation-aware task scheduling for heterogeneous multi-cloud systems. *Procedia Comput Sci* 50:176–184
35. Panda SK, Jana PK (2015) A multi-objective task scheduling algorithm for heterogeneous multi-cloud environment. In: *International Conference on Electronic Design, Computer Networks & Automated Verification (EDCAV)*, 2015, pp 82–87
36. Chunlin L, Jianhang T, Youlong L (2018) “Multi-queue scheduling of heterogeneous jobs in hybrid geo-distributed cloud environment. *J Supercomput* 74:5263–5292
37. Suri P, Rani S (2017) Design of task scheduling model for cloud applications in multi cloud environment. In: *International Conference on Information, Communication and Computing Technology*, pp 11–24
38. Grozev N, Buyya R (2013) Performance modelling and simulation of three-tier applications in cloud and multi-cloud environments. *Comput J* 58:1–22
39. Vieira CCA, Bittencourt LF, Madeira ERM (2015) A two-dimensional sla for services scheduling in multiple iaas cloud providers. *Int J Distribut Syst Technol* 6:45–64
40. Tsamoura E, Gounaris A, Tsihlias K (2013) Multi-objective optimization of data flows in a multi-cloud environment. In: *Proceedings of the Second Workshop on Data Analytics in the cloud*, pp 6–10
41. Miraftebzadeh SA, Rad P, Jamshidi M (2016) Efficient distributed algorithm for scheduling workload-aware jobs on multi-clouds. In: *11th System of Systems Engineering Conference (SoSE)*, 2016, pp 1–8
42. Moschakis IA, Karatza HD (2015) Multi-criteria scheduling of Bag-of-Tasks applications on heterogeneous interlinked clouds with simulated annealing. *J Syst Softw* 101:1–14
43. Panda SK, Jana PK (2016) Normalization-based task scheduling algorithms for heterogeneous multi-cloud environment. *Inf Syst Front* 20:373–399
44. Montes JD, Zou M, Singh R, Tao S, Parashar M (2014) Data-driven workflows in multi-cloud marketplaces. In: *IEEE 7th International Conference on Cloud Computing (CLOUD)*, 2014, pp 168–175
45. Bendoukha S, Bendoukha H, Moldt D (2015) ICNETS: Towards designing inter-cloud workflow management systems by petri nets. In: *Workshop on Enterprise and Organizational Modeling and Simulation*, 2015, pp 187–198
46. Mehdi NA, Holmes B, Mamat A, Subramaniam SK (2012) Sharing-aware intercloud scheduler for data-intensive jobs. In: *International Conference on Cloud Computing Technologies, Applications and Management (ICCTAM)*, 2012, pp 22–26
47. Abdi S, PourKarimi L, Ahmadi M, Zargari F (2017) Cost minimization for deadline-constrained bag-of-tasks applications in federated hybrid clouds. *Future Gener Comput Syst* 71:113–128
48. Gupta I, Kumar MS, Jana PK (2016) Transfer time-aware workflow scheduling for multi-cloud environment. In: *International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp 732–737
49. Gupta I, Kumar MS, Jana PK (2016) Compute-intensive workflow scheduling in multi-cloud environment. In: *International Conference on Advances in Computing, Communications and Informatics (ICACCI)* 2016, pp 315–321
50. Jrad F, Tao J, Streit A (2013) A broker-based framework for multi-cloud workflows. In: *Proceedings of the 2013 International Workshop on Multi-Cloud Applications and Federated Clouds*, pp 61–68
51. Lin B, Guo W, Chen G, Xiong N, Li R (2015) Cost-driven scheduling for deadline-constrained workflow on multi-clouds. In: *IEEE International Parallel and Distributed Processing Symposium Workshop (IPDPSW)*, 2015, pp 1191–1198
52. Hu H, Li Z, Hu H, Chen J, Ge J, Li C, Chang V (2018) Multi-objective scheduling for scientific workflow in multicloud environment. *J Netw Comput Appl* 114:108–122
53. Fard HM, Prodan R, Fahringer T (2013) A truthful dynamic workflow scheduling mechanism for commercial multicloud environments. *IEEE Trans Parallel Distrib Syst* 24:1203–1212
54. Chen W, Lee YC, Fekete A, Zomaya AY (2015) Adaptive multiple-workflow scheduling with task rearrangement. *J Supercomput* 71:1297–1317

55. Maheshwari K, Jung E-S, Meng J, Morozov V, Vishwanath V, Kettimuthu R (2016) Workflow performance improvement using model-based scheduling over multiple clusters and clouds. *Future Gener Comput Syst* 54:206–218
56. Sooezi N, Abrishami S, Lotfian M (2015) Scheduling data-driven workflows in multi-cloud environment. In: *IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2015, pp 163–167
57. Zhang J, Wang M, Luo J, Dong F, Zhang J (2015) Towards optimized scheduling for data-intensive scientific workflow in multiple datacenter environment. *Concurr Comput* 27:5606–5622
58. Panda SK, Jana PK (2015) Efficient task scheduling algorithms for heterogeneous multi-cloud environment. *J Supercomput* 71:1505–1533
59. Tejaswi TT, Azharuddin M, Jana PK (2015) A GA based approach for task scheduling in multi-cloud environment. *arXiv preprint arXiv:1511.08707*
60. Alsughayyir A, Erlebach T (2017) A Bi-objective Scheduling approach for energy optimisation of executing and transmitting HPC applications in decentralised multi-cloud systems. In: *16th International Symposium on Parallel and Distributed Computing (ISPDC)*, 2017, pp 44–53
61. Demchenko Y, Blanchet C, Loomis C, Branchat R, Slawik M, Zilci I, Bedri M, Gibrat J-F, Lodygensky O, Zivkovic M (2016) Cyclone: a platform for data intensive scientific applications in heterogeneous multi-cloud/multi-provider environment. In: *IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, 2016, pp 154–159
62. Frincu ME, Craciun C (2011) Multi-objective meta-heuristics for scheduling applications with high availability requirements and cost constraints in multi-cloud environments. In: *Fourth IEEE International Conference on Utility and Cloud Computing (UCC)*, 2011, pp 267–274
63. Geethanjali M, Sujana JAJ, Revathi T (2014) Ensuring truthfulness for scheduling multi-objective real time tasks in multi cloud environments. In: *International Conference on Recent Trends in Information Technology (ICRTIT)*, 2014, pp 1–7
64. Kang S, Veeravalli B, Aung KMM (2014) Scheduling multiple divisible loads in a multi-cloud system. In: *IEEE/ACM 7th International Conference on Utility and Cloud Computing (UCC)*, 2014, pp 371–378
65. Panda SK, Jana PK (2016) Uncertainty-based QoS min–min algorithm for heterogeneous multi-cloud environment. *Arab J Sci Eng* 41:3003–3025
66. Panda SK, Gupta I, Jana PK (2017) Task scheduling algorithms for multi-cloud systems: allocation-aware approach. *Inf Syst Front* 21:1–19
67. Panda SK, Pande SK, Das S (2018) Task partitioning scheduling algorithms for heterogeneous multi-cloud environment. *Arab J Sci Eng* 43:913–933
68. Jena T, Mohanty J (2017) GA-based customer-conscious resource allocation and task scheduling in multi-cloud computing. *Arab J Sci Eng* 43:1–16
69. Kang S, Veeravalli B, Aung KMM (2018) Dynamic scheduling strategy with efficient node availability prediction for handling divisible loads in multi-cloud systems. *J Parallel Distrib Comput* 113:1–16
70. Heilig L, Lalla-Ruiz E, Voß S (2016) A cloud brokerage approach for solving the resource management problem in multi-cloud environments. *Comput Ind Eng* 95:16–26
71. Uskenbayeva R, Kuandykov A, Cho Y, Kalpeyeva ZB (2014) Tasks scheduling and resource allocation in distributed cloud environments. In: *14th International Conference on Control, Automation and Systems (ICCAS)*, 2014, pp 1373–1376
72. Sandhu R, Sood SK (2015) Scheduling of big data applications on distributed cloud based on QoS parameters. *Cluster Comput* 18:817–828
73. Alsughayyir A, Erlebach T (2016) Energy aware scheduling of HPC tasks in decentralised cloud systems. In: *24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing (PDP)*, 2016, pp 617–621
74. Kessaci Y, Melab N, Talbi E-G (2013) A Pareto-based metaheuristic for scheduling HPC applications on a geographically distributed cloud federation. *Cluster Comput* 16:451–468
75. Garg SK, Yeo CS, Anandasivam A, Buyya R (2011) Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers. *J Parallel Distrib Comput* 71:732–749
76. Daochao H, Chunge Z, Hong Z, Xinran L (2014) Resource intensity aware job scheduling in a distributed cloud. *China Commun* 11:175–184
77. Yin L, Sun J, Zhao L, Cui C, Xiao J, Yu C (2015) Joint scheduling of data and computation in geo-distributed cloud systems. In: *15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2015, pp 657–666

78. Jing C, Zhu Y, Li M (2013) Customer satisfaction-aware scheduling for utility maximization on geo-distributed cloud data centers. In: IEEE 10th International Conference on High Performance Computing and Communications & 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC\_EUC), 2013, pp 218–225
79. Larsson L, Henriksson D, Elmroth E (2011) Scheduling and monitoring of internally structured services in cloud federations. In: IEEE Symposium on Computers and Communications (ISCC), 2011, pp 173–178
80. Yao MD, Chen DL, Chen X (2014) Scheduling system for cloud federation across multi-data center. *Appl Mech Mater* 457:839–843
81. Chudasama V, Shah J, Bhavsar M (2017) Weight based workflow scheduling in cloud federation. In: International Conference on Information and Communication Technology for Intelligent Systems, 2017, pp 405–411
82. Coutinho RDC, Drummond LM, Frota Y, de Oliveira D (2015) Optimizing virtual machine allocation for parallel scientific workflows in federated clouds. *Future Gener Comput Syst* 46:51–68
83. Durillo JJ, Prodan R, Barbosa JG (2015) Pareto tradeoff scheduling of workflows on federated commercial clouds. *Simul Model Pract Theory* 58:95–111
84. Nguyen P-D, Thoai N (2016) DrBCF: a differentiated ratio-based approach to job scheduling in cloud federation. In: 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), 2016, pp 31–37
85. Rubio-Montero A, Huedo E, Mayo-García R (2017) Scheduling multiple virtual environments in cloud federations for distributed calculations. *Future Gener Comput Syst* 74:90–103
86. Gouasmi T, Louati W, Kacem AH. Optimal MapReduce Job Scheduling algorithm across. *Cloud Federation*
87. Gouasmi T, Louati W, Kacem AH (2018) Exact and heuristic MapReduce scheduling algorithms for cloud federation. *Comput Electr Eng* 69:274–286
88. Gouasmi T, Louati W, Kacem AH (2017) Cost-efficient distributed MapReduce job scheduling across cloud federation. In: IEEE International Conference on Services Computing (SCC), 2017, pp 289–296
89. Kintsakis AM, Psomopoulos FE, Symeonidis AL, Mitkas PA (2017) Hermes: seamless delivery of containerized bioinformatics workflows in hybrid cloud (HTC) environments. *SoftwareX* 6:217–224
90. Van den Bossche R, Vanmechelen K, Broeckhove J (2013) Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Gener Comput Syst* 29:973–985
91. Van den Bossche R, Vanmechelen K, Broeckhove J (2011) Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds. In: IEEE Third International Conference on Cloud Computing Technology and Science (CloudCom), 2011, pp 320–327
92. Qiu X, Yeow WL, Wu C, Lau FC (2013) Cost-minimizing preemptive scheduling of mapreduce workloads on hybrid clouds. In: IEEE/ACM 21st International Symposium on Quality of Service (IWQoS), 2013, pp 1–6
93. Chopra N, Singh S (2013) HEFT based workflow scheduling algorithm for cost optimization within deadline in hybrid clouds. In: Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp 1–6
94. Duan R, Goh RSM, Zheng Q, Liu Y (2014) Scientific workflow partitioning and data flow optimization in hybrid clouds. *IEEE Trans Cloud Comput*
95. Duan R, Prodan R (2014) Cooperative scheduling of bag-of-tasks workflows on hybrid clouds. In: IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), 2014, pp 439–446
96. Kintsakis AM, Psomopoulos FE, Mitkas PA (2016) Data-aware optimization of bioinformatics workflows in hybrid clouds. *J Big Data* 3:20
97. Sharif S, Taheri J, Zomaya AY, Nepal S (2014) Online multiple workflow scheduling under privacy and deadline in hybrid cloud environment. In: IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom), 2014, pp 455–462
98. Chopra N, Singh S (2013) Deadline and cost based workflow scheduling in hybrid cloud. In: 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013, pp 840–846
99. Sharif S, Taheri J, Zomaya AY, Nepal S (2014) Online multiple workflow scheduling under privacy and deadline in hybrid cloud environment. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, 2014, pp 455–462

100. Chang YS, Fan CT, Sheu RK, Jhu SR, Yuan SM (2018) An agent-based workflow scheduling mechanism with deadline constraint on hybrid cloud environment. *Int J Appl Eng Res* 31:e3401
101. Luo H, Yan C, Hu ZJ (2015) An enhanced workflow scheduling strategy for deadline guarantee on hybrid grid/cloud infrastructure, vol 18, pp 67–78
102. Krishnan P, Aravindhar J (2019) Self-adaptive PSO memetic algorithm for multi objective workflow scheduling in hybrid cloud. *Int Arab J Inf Technol* 16:928–935
103. Marcon DS, Bittencourt LF, Dantas R, Neves MC, Madeira ER, Fernandes S, Kamienski CA, Barcelos MP, Gaspary LP, da Fonseca NL (2013) Workflow specification and scheduling with security constraints in hybrid clouds. In: 2nd IEEE Latin American Conference on Cloud Computing and Communications (LatinCloud), 2013, pp 29–34
104. Zhu J, Li X, Ruiz R, Xu X, Zhang Y (2016) Scheduling stochastic multi-stage jobs on elastic computing services in hybrid clouds. In: 2016 IEEE International Conference on Web Services (ICWS), 2016, pp 678–681
105. Chopra N, Singh S (2013) HEFT based workflow scheduling algorithm for cost optimization within deadline in hybrid clouds. In: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp 1–6
106. Duan R, Goh RSM, Zheng Q, Liu C (2014) Scientific workflow partitioning and data flow optimization in hybrid clouds. *IEEE Trans Cloud Comput* 1:1
107. Duan R, Prodan R (2014) Cooperative scheduling of bag-of-tasks workflows on hybrid clouds. In: 2014 IEEE 6th International Conference on Cloud Computing Technology and Science, pp 439–446
108. Bittencourt LF, Madeira ERM (2011) HCOC: a cost optimization algorithm for workflow scheduling in hybrid clouds. *J Int Serv Appl* 2:207–227
109. Bittencourt LF, Senna CR, Madeira ER (2010) Scheduling service workflows for cost optimization in hybrid clouds. In: International Conference on Network and Service Management (CNSM), 2010 pp 394–397
110. Lin B, Guo W, Lin X (2016) Online optimization scheduling for scientific workflows with deadline constraint on hybrid clouds. *Concurr Comput* 28:3079–3095
111. Zhang Y, Sun J (2017) Novel efficient particle swarm optimization algorithms for solving QoS-demanded bag-of-tasks scheduling problems with profit maximization on hybrid clouds. *Concurr Comput* 29:e4249
112. Wang W-J, Chang Y-S, Lo W-T, Lee Y-K (2013) Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments. *J Supercomput* 66:783–811
113. Jiang WZ, Sheng ZQ (2012) A new task scheduling algorithm in hybrid cloud environment. In: International Conference on Cloud and Service Computing (CSC), 2012, pp 45–49
114. Zhang Y, Sun J, Wu Z (2017) An heuristic for bag-of-tasks scheduling problems with resource demands and budget constraints to minimize makespan on hybrid clouds. In: Fifth International Conference on Advanced Cloud and Big Data (CBD), 2017, pp 39–44
115. Pelaez V, Campos A, Garcia DF, Entrialgo J (2016) Autonomic scheduling of deadline-constrained bag of tasks in hybrid clouds. In: International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS), pp 1–8
116. Sajid M, Raza Z (2016) Turnaround time minimization-based static scheduling model using task duplication for fine-grained parallel applications onto hybrid cloud environment. *IETE J Res* 62:402–414
117. Champati JP, Liang B (2015) One-restart algorithm for scheduling and offloading in a hybrid cloud. In: IEEE 23rd International Symposium on Quality of Service (IWQoS), 2015, pp 31–40
118. Chunlin L, Jianhang T, Youlong L (2017) Distributed QoS-aware scheduling optimization for resource-intensive mobile application in hybrid cloud. *Cluster Comput* 21:1–18
119. Xu B, Peng Z, Xiao F, Gates AM, Yu J-P (2015) Dynamic deployment of virtual machines in cloud computing using multi-objective optimization. *Soft Comput* 19:2265–2273
120. Zhang G, Zuo X (2013) Deadline constrained task scheduling based on standard-PSO in a hybrid cloud. In: International Conference in Swarm Intelligence, pp 200–209
121. Li C, Li L (2015) Efficient market strategy based optimal scheduling in hybrid cloud environments. *Wireless Pers Commun* 83:581–602
122. Zuo X, Zhang G, Tan W (2014) Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud. *IEEE Trans Autom Sci Eng* 11:564–573

123. Zhu J, Li X, Ruiz R, Xu X, Zhang Y (2016) Scheduling stochastic multi-stage jobs on elastic computing services in hybrid clouds. In: *IEEE International Conference on Web Services (ICWS)*, 2016, pp 678–681
124. Chunlin L, LaYuan L (2017) Optimal scheduling across public and private clouds in complex hybrid cloud environment. *Inf Syst Front* 19:1–12
125. Zhang P, Lin C, Li W, Ma X (2017) Long-term multi-objective task scheduling with diff-serv in hybrid clouds. In: *International Conference on Web Information Systems Engineering*, 2017, pp 243–258
126. Jiang WZ, Sheng ZQ (2012) A new task scheduling algorithm in hybrid cloud environment. In: *2012 International Conference on Cloud and Service Computing*, pp 45–49
127. Dhananjay Kumar, Kavitha B, Padmavathy M, Harshini B, Preethi E, Varalakshmi P (2016) Optimized particle swarm optimization based deadline constrained task scheduling in hybrid cloud. *ICTACT J Soft Comput* 6(2):1117–1122
128. Fan Y, Liang Q, Chen Y, Yan X, Hu C, Yao H, Liu C, Zeng D (2015) Executing time and cost-aware task scheduling in hybrid cloud using a modified DE algorithm. In *International Symposium on Computational Intelligence and Intelligent Systems*, pp 74–83
129. Balagoni Y, Rao RR (2016) A cost-effective SLA-aware scheduling for hybrid cloud environment. In: *IEEE International Conference on Computational Intelligence and Computing Research (ICCCIC)*, pp 1–7
130. Van den Bossche R, Vanmechelen K, Broeckhove J (2013) Online cost-efficient scheduling of deadline-constrained workloads on hybrid clouds. *Future Gener Comput Syst* 29:973–985
131. R. Van den Bossche, K. Vanmechelen, and J. Broeckhove, “Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds. In: *2011 IEEE Third International Conference on Cloud Computing Technology And Science*, 2011, pp. 320–327
132. Qiu X, Yeow WL, Wu C, Lau FC (2013) Cost-minimizing preemptive scheduling of mapreduce workloads on hybrid clouds. In: *2013 IEEE/ACM 21st International Symposium on Quality of Service (IWQoS)*, pp 1–6
133. Zhang Y, Sun JJC (2017) Novel efficient particle swarm optimization algorithms for solving QoS-demanded bag-of-tasks scheduling problems with profit maximization on hybrid clouds. *Pract Exp* 29:e4249
134. Wang W-J, Chang Y-S, Lo W-T, Lee Y-K (2013) Adaptive scheduling for parallel tasks with QoS satisfaction for hybrid cloud environments. *J Supercomput* 66:783–811
135. Pelaez V, Campos A, Garcia DF, Entrialgo J (2016) Autonomic scheduling of deadline-constrained bag of tasks in hybrid clouds. In: *2016 International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS)*, pp 1–8
136. Champati JP, Liang B (2015) One-restart algorithm for scheduling and offloading in a hybrid cloud. In: *2015 IEEE 23rd International Symposium on Quality of Service (IWQoS)*, pp 31–40
137. Chunlin L, LaYuan L (2017) Optimal scheduling across public and private clouds in complex hybrid cloud environment. *Inf Syst Front* 19:1–12
138. Calheiros RN, Buyya R (2012) Cost-effective provisioning and scheduling of deadline-constrained applications in hybrid clouds. In: *International Conference on Web Information Systems Engineering*, pp 171–184
139. Yuan H, Bi J, Tan W, Li BH (2017) Temporal task scheduling with constrained service delay for profit maximization in hybrid clouds. *IEEE Trans Autom Sci Eng* 14:337–348
140. Yuan H, Bi J, Tan W, Zhou M, Li BH, Li J (2017) TTSA: an effective scheduling approach for delay bounded tasks in hybrid clouds. *IEEE Trans Cybern* 47:3658–3668