

# A Comprehensive Investigation of Workflow Scheduling in Cloud Computing Environment



Nagresh Kumar and Sanjay Kumar Sharma

**Abstract** The cloud resource scalability feature allows the users' applications to meet its need during runtime or before execution and hence it required to be organized dynamically as needed. These scalable and distributed features of the cloud resources allow workflow management systems to meet the expectation of the service provider and customer. The service-level agreement (SLA) is a major concern in workflow algorithms. It also looks at the economic benefits for service providers and customers. Due to these multi-objective natures of workflow scheduling and various constraints imposed by user and cloud environment, a large number of algorithms are suggested by various researchers. There is not a single algorithm proposed by researchers which handle all known constraint imposed by user and service provider. At the end of this research paper, the authors have suggested that workflow scheduling for cloud environments is an optimization problem.

**Keywords** Service-level agreement (SLA) · Workflow · Multi-objective

## 1 Introduction

Workflow is a sequence of a process that together accomplishes a task. The workflow is responsible to distribute the task at appropriate computing resources for effective and fast execution. A large number of workflow scheduling algorithms in the cloud environment have been proposed by the researcher with various considerations like resource constraint, homogeneous, and heterogeneous clustering, task dependencies, and utility-based computing. The workflow management system allows shared resources based on demand [1]. This is done by virtualizing the resources as per the required number of storage, CPU, and memory bandwidth. The VM can be leased or released as per requirement and charged for the unit billing period.

---

N. Kumar (✉) · S. K. Sharma

Department of Computer Science, Banasthali Vidyapith, Vanasthali, Rajasthan, India

## 2 Workflow Scheduling in Cloud Computing

Now, workflow scheduling in a cloud computing environment is a mechanism that involves in managing allocation to resources and preservation of execution of an interdependent task. Workflow implementation in a cloud environment requires two mechanisms: (i) resource allocation and (ii) resource provision. Resource provisioning is the selection and provision of computational resources used to execute tasks. This means that there are heuristics that can determine how many VMs are being rented, when execution starts and stop, their type of VM, etc. The second approach is about the mapping of tasks with appropriate resources. This process is called resource provisioning.

### 2.1 Cloud Workflow Management System

The responsibilities of a workflow management system are to supervise, build, and manage data and task transparently between distributed resources [2]. The workflow management system consists of various components to address tasks, data, and resource concerns for customer quality of service requirements. The reference architecture shown in Fig. 1 can be classified into three parts.

- (A) **User interface**—The user interface allows the user to work with workflow creation, execution, submission, and workflow monitoring. All these are web services and distributed through a web portal or standalone application at the user end. Users are responsible for defining task attributes and linking them based on their data dependencies.
- (B) **Workflow engine**—The core part of the workflow management system is the workflow engine. It is responsible for managing the workflow. The parser modules of the workflow engine are responsible for translating high-level workflow descriptions for tasks and data objects. The scheduling components are responsible for implementing user-defined scheduling policies and giving space to the workflow at various stages of implementation. Works with data dispatcher and task resource interface plugins are constantly monitors tasks in the workflow.
- (C) **Plug-in**—Plug-in of WFMS is responsible for communication between various resource management systems for distribution of resources. Plugins are responsible for querying work and data attributes, transferring data from and to resources and monitoring the execution status of tasks and applications to run, and recover working status, and also to measure power consumption. three parts.

The bottom part of architecture includes cloud, grid, and cluster. Further resource managers are responsible for communication with market resources, scalable application managers, and various cloud services for resource management [3].

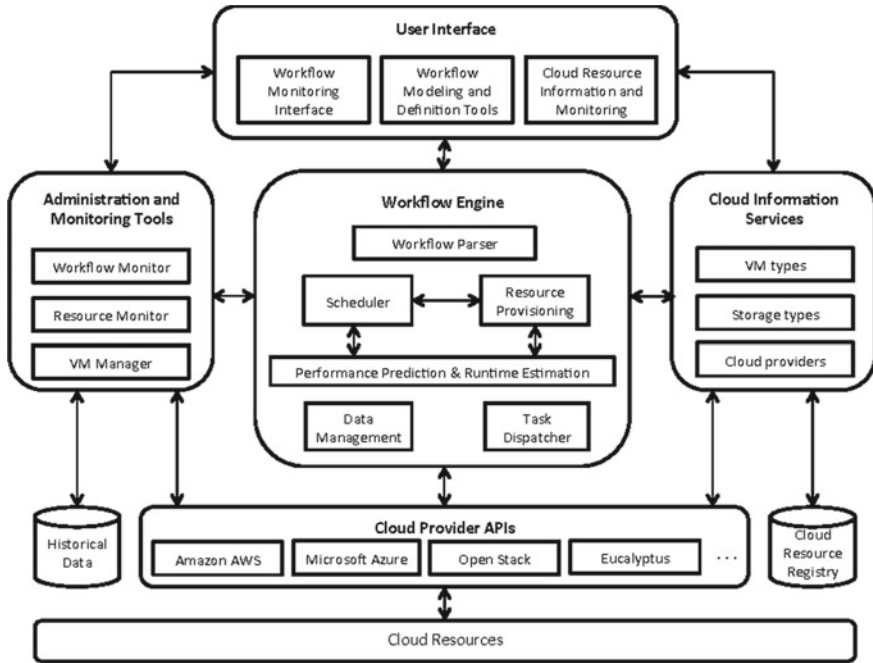


Fig. 1 Cloud workflow architecture [7]

## 2.2 Goals of the Workflow Scheduling Algorithm

In general, cost optimization is a common goal of workflow scheduling algorithms. Other performance metrics such as overall runtime, power consumption, safety, maximization of workload, VM utilization, awareness, security awareness, and reliability are also major concerns of workflow scheduling algorithms [4].

**Resource provisioning**—Allocating tasks to appropriate resources are an important challenge for workflow algorithms. The research work is shown by Michan et al. [5], and Villegas et al. [6] researched on the dependency between task and resources.

**Performance variation and other sources of uncertainty**—Cloud features like virtualization, resource sharing nature, and diversity of non-virtualized hardware in the cloud are the result of diversity in resource performance [8–12]. Due to uncertainty among the dependencies in a large workflow schedule cannot make an accurate decision about runtime and QoS requirements [13–15].

**Utility-based pricing model**—Some schedulers are interested in utility-based pricing model. For example, Amazon’s EC2 Spot Instance feature is available for dynamic pricing. Spot instances vary over time based on market price and demand [16].

### 3 Related Works

This section has taken a detail surveys on various workflow scheduling algorithms:

- A. **Deadline and energy-aware task scheduling (DEATS) in cloud**—In this paper, the authors proposed the task scheduling algorithm concern to energy-aware and deadline aware. The proposed algorithm is to reduce the makespan and reduce energy consumption in meeting the time limits. Final results in this research paper show that it is research performance over FCFS and EDF. The authors also suggest integrating other optimization methods to improve performance [17].
- B. **Optimizing power consumption for expiration-constrained workflow in cloud data centers**—In this research paper, the authors take a survey about electricity consumption by datacenter. The authors proposed TMMS and WTM adopted scheduling algorithms to optimize power consumption. This proposed algorithm outperforms better in terms of the number of VM used, electricity consumed by the data center. In this proposed work, electricity cost, power consumption model, and time taken inside and outside the VM are not taken into account [18].
- C. **Cost-effective fault-tolerant (CEFT) scheduling algorithms for real-time tasks in cloud systems**—In this paper, the authors proposed a novel CEFT algorithm to reduce implementation costs while trying to do more to meet the expiration requirements. The CEFT algorithm optimizes repetition and resource allocation in a repetitive manner with PSO. The rescheduling method is followed to increase the time limit guarantee ratio. Simulation results show that CEFT works better in terms of guarantee ratio and average cost, which ensures the feasibility and effectiveness of CEFT [19, 20].
- D. **Cost- and energy-efficient algorithm for scientific workflow with the time limit in the clouds**—In this research paper, the authors proposed an algorithm that is cost-effective and energy-efficient. The authors proposed five steps to meet the deadline and reduce electricity consumption. All these steps effectively reduce the cost of workflow. Experimental results have shown that it performs better in terms of cost and power consumption. In future work, the author suggests taking into account the actual power cost for workflow scheduling [21, 22].
- E. **An improved genetic algorithm to solve scheduled work problems with time limits on hybrid clouds**—In this research paper, the authors proposed a modified version of GA using novel crossover which identifies the world's best genes hidden in the population to inherit offspring. The authors experimentally shown that it performs better than normal GA [23].
- F. **Multi-objective hybrid cloud resource scheduling method and scheduling method based on expiration and cost limits**—In this paper, the authors developed a mosaic algorithm on hybrid cloud computing by combining two single

objective optimization algorithms. These single objective optimization algorithms used improved ant colony algorithm by following entropy optimization. Proposed algorithm optimizing resource utilization subject to cost limits and time duration with the objective of resource utilization cost, QoS, task completion time, and service provider interest [24].

- G. **Better budget-time-scheduling algorithms to work on constrained resources**—In this paper, the authors improve the scheduling algorithm for a variety of systems. The schedule is less focused and less complicated to meet budget and deadline limits. The algorithm is compared to three recent algorithms, such as the DBCS, BHEFT, and HBCS algorithms. Experimental result shows a high success rate when time and cost constraints are tight. In the future, author wants to minimize makespan according to the condition that the total cost of the workflow [25, 26].
- H. **Multi-objective accelerated particle swarm optimization technique for scientific workflow in the IaaS cloud**—In this research paper, the authors proposed an algorithm that works in two steps. First, the algorithm follows a strategy that finds the most suitable VM which reduces the execution time. Second, the algorithm creates a proper schedule of tasks that can increase the output of the server. Through several performance analyses, the authors demonstrated the dominance of the proposed algorithm over sophisticated algorithms [27, 28].
- I. **Particle swarm optimization-based resource provisioning and scheduling algorithm**—In this paper, the author proposed a particle swarm optimization-based resource provisioning cost minimization which considers resource diversity, VM performance, and elastic provisioning. Scheduling and resource provisioning are integrated to resolve PSO issues. The advantage of the global optimization technique algorithm is that it permits the creation of good quality schedules. The authors have pointed out that as unlimited resources are successfully extracted, the calculation overhead increases rapidly in the types of workflows and VMs provided by the provider [29].
- J. **Budget-driven scheduling of scientific workflows in IaaS clouds with fine-grained billing periods**—In this paper; the authors present task-based budget delivery techniques to implement scientific workflows to reduce makespan. The proposed algorithm divides the workflow budget and guides the resource utilization of the divided budget of each task. The proposed algorithm implements the VM recharge policy to use idle time slots caused by a rigorous and fast billing period. Since reusable passive VMs are not available; it provides the fastest VM type on a budget when needed [30].

**Fault-tolerant schedule using spot instances**—Authors in this research paper proposed an algorithm which works on on-demand VM and spot VM. According to the proposed algorithm, on-demand VM will change according to the user-defined time limit. The author also proposed a bidding strategy for spot VM [31].

## 4 Basic Elements for Workflow Scheduling Algorithms

On the basis of a survey taken in Sect. 3, this section describes the various approaches to developing workflow algorithms in IaaS cloud environments.

### 4.1 Task-VM Mapping Dynamics

The classification of the scheduling algorithms generally proposed by Casavent and Kuhl [32], and categories as static and dynamic. In addition to these two cloud environments, it can combine static and dynamic environments to get the most out of it, i.e., hybrid.

**Static**—This type of scheduling algorithm produces advanced task-VM mapping and does not change during the run. This perspective is not allowed to be dynamic and therefore not subject to change. A small change in Task-VM mapping can lead to implementation failure and therefore affect QoS.

**Dynamic**—Dynamic scheduling algorithm determines task resource mapping during work. These runtime decisions are based on workflow implementation and the latest system status. Dynamic scheduling algorithm decisions are taken during the run when ready to execute. These regulations allow us to adapt to changes in the environment.

**Hybrid**—In cloud environments, both static node dynamic scheduling has its own significance. To take advantage of these, a new algorithm must be designed to find the difference between the compatibility and performance of the dynamic scheduling algorithm. To build a hybrid algorithm for categories into two parts: Runtime Purification and Sub-Workflow Static.

### 4.2 Resource Allocation Strategy

Kindly, resource allocation is categorized as static and dynamic resource allocation. In a static resource allocation, all resources are configured before implementation in the workflow. In dynamic resource allocation, all decisions are made during runtime.

**Static VM pool**—This strategy can be used through static resource allocation. When the VM pool is split, the resources are rented and become active during workflow execution. Its implementation resources will be released when the workflow is completed. The advantages of such an algorithm are that the algorithm mainly focuses on the task VM allocation after making resource allocation decisions.

**Elastic VM pool**—The workflow algorithm in this category follows static and dynamic resource allocation strategies. This mechanism permits the algorithm to change the type of VM and the number of VM during runtime. Due to flexibility on

VM, the workflow algorithm in these categories may take a decision on cost, number of VM used, and output.

### 4.3 Optimization Strategy Classification

The most important part of the optimization strategy algorithm is the mathematical model of problem and solution space. The optimization technique is classified into two categories, i.e. optimal and suboptimal, in which suboptimal is classified into three categories.

**Heuristic**—Heuristic is the property of rules for the solution of a particular problem. These types of rules are specific and designed for a specific problem to find solution within an acceptable time. The main purpose of heuristic-based scheduling algorithms is to find satisfactory solutions in a timely manner.

**Meta-heuristic**—Meta-heuristic algorithms are commonly used for optimal solutions to a problem. In general meta-heuristic algorithms are more computationally intensive and take longer to solve workflows using meta-heuristic algorithms, which is a challenging problem due to the virtualization of an unlimited number of resources and dependencies [33].

**Hybrid**-Meta heuristic algorithms are used to solve a set of tasks in the workflow and heuristic algorithms are used to solve the task in a small simplified workflow. In such an approach, we get better optimization than heuristic and meta-heuristic algorithms.

### 4.4 Resource Model Classification

This section presents a resource model based on the assumption of the classification algorithm. Classification is done with services provided by providers like VM pricing model and data transfer cost. Resource model is classified as follows-

#### 4.4.1 VM Leasing Model

This model is maintained by assuming that the classification service providers provide restricted or unrestricted leased VMs for a given customer.

**Restricted**—Limited VM leasing model service provider has a high limit on the number of virtual machines that a customer can lease. Algorithms that use this model work with a limited number of VM resources.

**Unrestricted**—In these VMs leasing model algorithm assumes that there is no limit on the number of VM to lease to a customer.

#### 4.4.2 Deployment Model

The workflow algorithm under the deployment model is classified on the basis of the datacenter and resources leased by the service provider.

**Single service provider**—The algorithm in this approach considers a single cloud service provider for IaaS. There is no need to consider inside and outside data transfer cost.

**Multiple service providers**—In this category, algorithms consider different cloud service providers to schedule tasks. The different service providers may have different service level agreements and different resource provisioning price offerings. Now it's the scheduler's choice to choose suitable offers.

**Single data center**—These taxonomic algorithms choose to provide VMs in a single data center. This approach reduces latency and faster transfer time and costs because most providers do not charge for transfers made in the data center.

**Multiple Data Centers**—In this classification of workflow algorithms, service provider offers VM available in different datacenter which is geographically distributed. It is most suitable for the application whose inputs are from various geographical locations. It reduces the latency and data transfer cost.

#### 4.4.3 Data Sharing Model

Data sharing in the workflow process affects the performance of the scheduling algorithm because it affects scaling, cost, makespan, etc. There are two common approaches are used to improve the performance of the scheduling algorithm—(i) peer-to-peer and (ii) shared storage.

**P2P**—In this process, data are transferred directly from the VM running child process to the VM running parent process. This mechanism leads to synchronize the communication. This process leads to higher cost because lease time increased but increase the scalability.

**Shared storage**—This process reduces the VM leading cost and improves the resource utilization by storing the output in globally shared storage systems like AWS simple storage system (S3) or AWS elastic file system (EFS). There are many advantages of such a system. First, data integrity is maintained and recovered data in case of failure. Second, it permits asynchronous computing because data can be released to a VM running an actual function while continuing in the storage system.

#### 4.4.4 Data Transfer Cost Awareness

The data transfer pricing plan is different in a different scenario. Trans bound and inbound data are usually independent, and transferring data from a cloud provider is charged and expensive. In many service providers, algorithms that schedule workflows have to relate to this cost because data transfer may belong to different service provider.



#### 4.4.5 Storage Cost Awareness

In this strategy, an algorithm decides the cost of storage based on the type of operation stored, as well as the amount of data stored. The algorithm for storage, therefore, agrees with the cost and generally estimates the data size and data based on a fixed price per unit.

#### 4.4.6 VM Price Model

In this category, four different price models are proposed as static, dynamic, subscription based, and time unit.

#### 4.4.7 VMcore Calculation

This classification refers to algorithms that schedule multiple tasks simultaneously about multiple VMs.

**Single core**—In this mechanism, algorithm assumes that the VM has only one core. It simplifies the scheduling process because it resolves resource conflicts.

**Multi-core**—In the multicore facility, IaaS provides multiple cores. Workflow scheduling takes advantage of this task to execute multiple tasks simultaneously on a single VM, possibly avoiding the time, expense, and intermediate data transfers.

## 5 Analysis

Workflow scheduling in cloud computing is a very challenging task due to various constraints and multi-objective nature. So there is a need for an optimization technique that works on multi-objective nature and constraints imposed by the user and cloud environment.

Let  $T_n$  is the task with  $n$  objective then we need to optimize the objective function-  $T_n(X)$ , where  $X$  is a variable and  $X = 1, 2, 3, \dots, N$ , constraints to.

$F_m(X)$  where  $m = 1, 2, \dots, M$

$F$  is denoting constraints function,  $m$  is the number of constraints, and  $X$  is the domain of feasible solution. Suppose problem A is decision problem, and problem B is optimization problem; then it is possible that  $A \alpha B$ . Hence, we can say that such type of problem can be solved only by multi-objective optimization technique. A diagram given in Fig. 2 completely described that how a multi-objective workflow scheduling algorithm can be incorporated.

On the basis of various taxonomy in Sect. 4 and survey in Sect. 3, we have found that such scenario can be handled by a multi-objective optimization technique

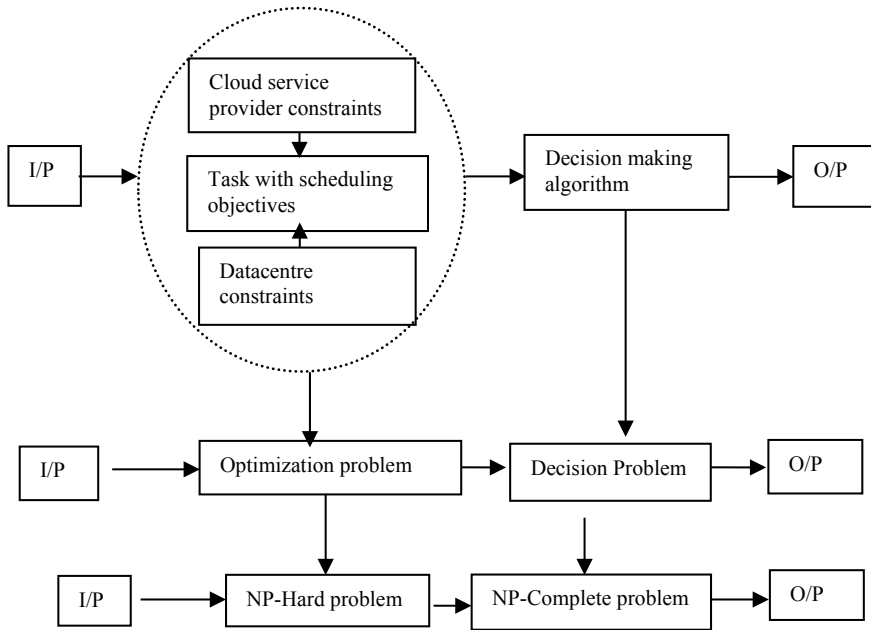


Fig. 2 Workflow is an NP-problem

like genetic algorithm, ant colony optimization algorithm, particle swarm optimization algorithm, etc. Genetic algorithm and ant colony optimization technique are mathematically well proof than PSO but PSO works with very few parameters [34].

## 6 Conclusion

Workflow scheduling in cloud computing is a very challenging task due to various constraints imposed during service-level agreements (SLAs) by users and the cloud environment. These various constraints make workflow scheduling as a multi-objective nature. A multi-objective problem can be solved only with optimization techniques with a decision problem. Since it is an NP-hard problem so there is a need for a meta-heuristic optimization technique that works suitably in such type of environment. As a future scope, multi-objective optimization techniques which address a maximum number of constraints should be implemented and compared.

## References

1. Buyya R, Yeo CS, Venugopal S, Broberg J, Brandic I (2009) Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generat Comput Syst* 25(6):599–616. ISSN 0167-739X
2. Yu J, Buyya R (2004) A novel architecture for realizing grid workflow using tuple spaces. In: *Proceedings of the fifth IEEE/ACM international workshop on grid computing*. IEEE, Pittsburgh, USA, pp 119–128
3. Vöckler J, Juve G, Deelman E, Rynge M, Berriman BG (2011) Experiences using cloud computing for a scientific workflow application. In: *Proceedings of 2nd workshop on scientific cloud computing (ScienceCloud 2011)*
4. Zeng L, Veeravalli B, Li X (2015) Saba: a security-aware and budget-aware workflow scheduling strategy in clouds. *J Parallel Distrib Comput* 75:141–151
5. Michon E, Gossa J, Genaud S et al (2012) Free elasticity and free CPU power for scientific workloads on IaaS clouds. In: *Proceedings of the eighteen IEEE international conference on parallel and distributed systems (ICPADS)*. IEEE, Singapore, pp 85–92
6. Villegas D, Antoniou A, Sadjadi SM, Iosup A (2012) An analysis of provisioning and allocation policies for infrastructure-as-a-service clouds. In: *12th IEEE/ACM international symposium on cluster, cloud and grid computing*, pp 612–619, 13–16 May 2012. ISBN: 978-1-4673-1395-7
7. Rodríguez M, Buyya R (2016) A taxonomy and survey on scheduling algorithms for scientific workflows in IaaS cloud computing environments: workflow scheduling algorithms for clouds. *Concurrency Comput Pract Exp* 29(8). ISSN: 1532-0626
8. Schad J, Dittrich J, Quiané-Ruiz JA (2010) Runtime measurements in the cloud: observing, analyzing, and reducing variance. In: *36th International conference on very large data bases, Singapore. Proc VLDB Endowment* 3(1–2):460–471, 13–17 Sept 2010
9. Ostermann S, Iosup A, Yigitbasi N, Prodan R, Fahringer T, Epema D (2010) A performance analysis of EC2 cloud computing services for scientific computing. In: *Cloud Computing*. Springer, Munich, Germany, pp 115–131
10. Gupta A, Milojicic D (2011) Evaluation of HPC applications on cloud. In: *Open cirrus summit (OCS), 2011 Sixth, Atlanta, Georgia*, pp 22–26
11. Iosup A, Ostermann S, Yigitbasi MN, Prodan R, Fahringer T, Epema D (2011) Performance analysis of cloud computing services for many-tasks scientific computing. *IEEE Trans Parallel Distrib Syst* 22(6):931–945
12. Jackson KR et al (2010) Performance analysis of high performance computing applications on the amazon web services cloud. In: *2010 IEEE second international conference on cloud computing technology and science*, pp 159–168. ISBN: 978-1-4244-9405-7
13. Nabrzyski J, Schopf JM, Weglarz J (2012) *Grid resource management: state of the art and future trends*, vol 64. Springer Science & BusinessMedia, Berlin, Germany
14. Valentin C, Ciprian D, Corina S, Florin P, Alexandru C (2010) *Large-scale distributed computing and applications: models and trends*
15. Berman F, Fox G, Hey Anthony JG (2003) *Grid computing: making the global infrastructure a reality*, vol 2. John Wiley and sons, Hoboken, New Jersey, United States
16. Amazon EC2 Spot Instances. Available on: <https://aws.amazon.com/ec2/spot/?cards.sort-by=item.additionalFields.startDateTime&cards.sort-order=asc/>. Accessed on 15 Oct 2020
17. Hicham BE, Said BE, Touhafi A, Ezzati A (2018) Deadline and energy aware task scheduling in cloud computing. In: *4th International conference on cloud computing technologies and applications (Cloudtech)*, 26–28 Nov 2018. ISBN: 978-1-7281-1637-2
18. Al-Dulaimy A, Itani W, Zekri A et al (2016) Power management in virtualized data centers: state of the art. *J Cloud Comput* 5:6
19. Guo P, Liu M, Wu J, Xue Z, He X (2018) Energy-efficient fault-tolerant scheduling algorithm for real-time tasks in cloud-based 5G networks. *IEEE Access*, pp 1–1. <https://doi.org/10.1109/ACCESS.2018.2871821>
20. Wu L, Ding R, Jia Z, Li X (2020) Cost-effective resource provisioning for real-time workflow in cloud. *Complexity* 2020(Article ID 1467274):15

21. Peng K, Zhao B, Xue S, Huang Q (2020) Energy- and resource-aware computation offloading for complex tasks in edge environment. *Complexity* 2020(Article ID 9548262):14
22. Zhu M, Wu Q, Zhao Y (2012) A cost-effective scheduling algorithm for scientific workflows in cloud. In: Proceedings of 31st IEEE international performance computing and communications conference
23. Yassa S, Sublime J, Chelouah R, Kadima H, Jo GS, Granado B (2013) A genetic algorithm for multi-objective optimisation in workflow scheduling with hard constraints. *Int J Metaheuristics* 2(4):415–433. <https://doi.org/10.1504/IJMHEUR.2013.058475>
24. Zuo L, Shu L, Dong S, Chen Y, Yan L (2017) A multi-objective hybrid cloud resource scheduling method based on deadline and cost constraints. *IEEE Access* 5:22067–22080
25. Sun T, Xiao C, Xu X, Tian G (2017) An improved budget-deadline constrained workflow scheduling algorithm on heterogeneous resources. In: 2017 IEEE 4th international conference on cyber security and cloud computing (CSCloud), New York, NY, pp 40–45. <https://doi.org/10.1109/CSCloud.2017.8>
26. Kaur N, Singh S (2016) A budget-constrained time and reliability optimization BAT algorithm for scheduling workflow applications in clouds. *Procedia Comput Sci* 98:199–204. ISSN 1877-0509
27. Konjaang JK, Xu L (2021) Multi-objective workflow optimization strategy (MOWOS) for cloud computing. *J Cloud Comput* 10:11
28. Adhikari M, Amgoth T (2018) Multi-objective accelerated particle swarm optimization technique for scientific workflows in IaaS cloud. In: 2018 International conference on advances in computing, communications and informatics (ICACCI), Bangalore, India, pp 1448–1454
29. Gill SS, Buyya R, Chana I et al (2018) BULLET: particle swarm optimization based scheduling technique for provisioned cloud resources. *J Netw Syst Manage* 26:361–400
30. Rodriguez MA, Buyya R (2017) Budget-driven scheduling of scientific workflows in IaaS clouds with fine-grained billing periods. *ACM Trans Auton Adapt Syst* 12(2, Article 5):22
31. Poola D, Ramamohana Rao K, Buyya R (2014) Fault-tolerant workflow scheduling using spot instances on clouds. *Procedia Comput Sci* 29:523–533
32. Casavant TL, Kuhl JG (1998) Taxonomy of scheduling in general-purpose distributed computing systems. *IEEE Trans Softw Eng* 14(2):141–154
33. Talbi EG (2009) *Metaheuristics: from design to implementation*, vol 74. Wiley & Sons, Hoboken, New Jersey, United States
34. Kumar N, Kumar Sharma S (2018) Inertia weight controlled PSO for task scheduling in cloud computing. In: 2018 International conference on computing, power and communication technologies (GUCON), pp 155–160. <https://doi.org/10.1109/GUCON.2018.867499>.