Workflow Scheduling In Cloud Computing–Models, Objectives, Recent Developments and Future Directions

Deafallah Alsadie

Department of Information Systems
Umm Al-Qura University
Makkah, Saudi Arabia

Abstract

Workflow scheduling is a significant challenge in a cloud computing environment that focuses on executing workflows as per the quality of service requirements such as execution cost and execution time. Several workflow scheduling methods have been developed in the recent past to schedule different types of workflows in cloud computing effectively. This paper introduces workflow scheduling, its types and objectives, followed by a comprehensive review of the recent development in workflow scheduling of cloud computing. It provides a taxonomy of workflows analyses the scheduling objectives. Different scheduling methods have been described and compared in multiple dimensions such as approach, scheduling type and scheduling objective. Finally, it highlights the promising directions to carry out future research in this field.

Keywords: *Cloud computing, Scientific workflows, scheduling, Resource management.*

1. Introduction

Cloud computing offers a school of computing resources connected through the Internet and provides many computing platforms and services in a distributed manner [1-5]. It provides different services and computing resources like memory, processing power, storage and network bandwidth with the minimum human intervention and other overheads [6-8]. Cloud computing provides three different types of services, platform as a service, software as a service and infrastructure as a service.

In the platform as a service model, cloud computing offers cloud users a platform that includes databases, operating systems, web server, and different programming languages. Cloud users can develop and deploy custom applications in the cloud computing platform without handling Software and Hardware required to develop applications. The most common examples include Amazon elastic beanstalk, Microsoft Windows Azure compute, Google App Engine, force.com etc. [9]. In the case of software as a service computing model, cloud computing provides different software as a service in the virtual desktop. Cloud users can request different application software as a service. The most common examples include Dropbox [38], Google apps, salesforce.com [10].

Cloud computing environment provides different computing hardware level resources using infrastructure as a Service model. Different computing resources are assigned and provisioned from Cloud users through virtual machines. The most common example includes Microsoft Azure [11] and Amazon ec2 [12]. In a cloud computing environment, cloud users are charged as pay as you use model basis. The computing resources can be assigned and provisioned depending upon the requirements of the cloud users without any additional system overhead. Cloud computing enables the maximization of revenue about computing resource performance and the cloud data centre's costs.

Recently, with the emergence of the Internet of things (IoT), enormous amounts of data have been generated and transmitted to the cloud for processing. Thus, cloud computing has become a primary paradigm for distributed computing because of its processing and storage capabilities. However, for real time applications that are latency sensitive, a cloud computing environment may increase response time and latency time, leading to wastage of computing resources and more power consumption of computing servers. In order to address the issue of latency in time sensitive applications, the concept of fog computing has been introduced. Fog computing provides processing of critical tasks on the network address using local fog devices instead of transmitting to cloud computing resources. Fog computing environment reaps the benefits of end user devices installed at network edges resulting in the execution of the specific tasks at the network edge. However, fog computing devices suffer from limited computing capacity and storage capacity limitations.

Therefore, both fog computing and cloud computing cannot efficiently meet the challenge mentioned above due to their advantages and disadvantages. In order to meet these challenges, a hierarchical model of fog computing environment has been introduced for real time applications. In this model, different applications are represented in the form of workflows that can be executed in a fog cloud computing model instead of a centralized cloud computing model. Formally, a workflow is represented as a directed acyclic graph, G (Ts, Eg), where Ts represents a set of tasks as $\{t_1, t_2. \ldots\ldots t_n\}$ [13]. These tasks give the task of the application with specific computing requirements for the computational load. Eg represents a list of graph edges defining inter task data dependencies. The cloud service provider receives the workload of different applications in the form of workflows [1]. The workflows represent scientific and engineering applications. Workflow execution requires computing resources to meet the quality of service requirement and specified time limits.

However, due to the availability of a huge number of applications in the cloud computing environment, the workflows require effective and efficient scheduling to meet the required quality of service and time limits as per the service level agreement. Due to differences in workflow requirements of computing resources, there may be an imbalance of computing resource management and turnaround time of different workflows.

Manuscript received March 5, 2022
Manuscript revised March 20, 2022
https://doi.org/10.22937/IJCSNS.2022.22.3.88

Workflow scheduling can allow service providers to execute different workflows within budget and deadline constraints. Workflow scheduling maps of workflow to appropriate computing resources for minimizing its cost and execution time by improving resource utilization under the constraints of service level agreement.

Several workflow scheduling methods have been developed in the cloud computing environment. These methods focus on optimizing objectives such as budget, reliability, availability, minimizing makespan, service level agreement violations, security and load balancing etc. Different researchers have divided workflow scheduling methods into different categories: static vs dynamic scheduling, centralized vs decentralized scheduling, preemptive versus non preemptive scheduling, online vs offline scheduling, and task level scheduling. Several research efforts have been invested in developing different types of workflow scheduling for optimizing different objectives as described above.

This paper provides a comprehensive review of currently existing methods for workflow scheduling in the cloud computing environment. It provides an overview of different types of scheduling, scheduling objectives and a comprehensive analysis of workflow scheduling methods in the cloud computing environment. This paper mainly contributes in the following ways.

- Introduces cloud workflow and its scheduling methods
- Presents a comprehensive review of workflow scheduling methods in the cloud computing environment
- Summarizes and highlights critical issues of workflow scheduling in cloud computing.
- Highlights primary future directions for developing methods for scheduling workflows in cloud computing for fellow researchers.

The rest of the article is structured as follows. Section 2 presents workflow scheduling model types. Section 3 highlights scheduling types. Section 4 presents workflow scheduling objectives. Section 5 analyzes workflow scheduling methods. Section 6 discusses the scheduling methods. Finally, section 7 concludes the paper at the end.

2. Workflow Scheduling Model Types

Literature has proposed many models for workflow scheduling in the cloud computing environment. These models can be divided into two categories, synthetic workflows, and scientific workflows. Each model contains some models representing different types of workflows generated by different applications. Generally, synthetic workflows are generated using some algorithm and primarily used to test different workflow scheduling methods in cloud computing environments. In contrast, scientific workflows are generated through different large scale scientific applications in a cloud computing environment. Different types of workflow scheduling models have been described below.

2.1 Synthetic workflows

Synthetic workflows can be categorized into the following categories based on task execution flow. The major categories include [14, 15].

- Parallel workflow
- Rules Driven workflow
- Sequential workflow
- State Machine workflow

The sequential workflow model consists of different tasks that depend upon the predecessor task. Execution of sequential workflow always moves forward without any backward path. In contrast, parallel workflow contains the execution of multiple tasks in parallel independent of the execution of other tasks. In the case of the state machine workflow model, there is a communication predecessor task to the next as per requirement. Such workflow is analogous to solving eight puzzle problems. Finally, the rule-driven workflow model consists of the execution of sequential workflow with certain constraints or rules controlling the execution process of the task. Workflow gets executed based upon specific criteria. Rule driven workflow model is also called immediate workflow.

2.2 Scientific workflows

Scientific workflows are generally generated using different large scale applications in the cloud computing environment. Workflow models also contain different types of structures and computational features [16-19]. This workflow may require different computing resources such as processor, memory and input output devices depending upon the requirements of the user applications [20]. The scientific workflow can be CPU intensive, memory intensive aur input-output intensive, requiring respective computing resources more compared to other workflows. For example, CPU intensive workflows require most CPU time for their execution. Whereas input-output intensive workflows spend most of their time performing input output operations. As described below, scientific workflows can be divided into five classes based on their characteristics.

- Montage workflow: this category of scientific workflow is defined based upon the astronomy concepts. Montage workflow uses the concept of astronomy involving a set of images for creating custom mosaics of the sky [22]. It computes the geometry of the projected mosaic and redesigns floods in the images. It involves generating a background radiation model for achieving moasoic flux scale and background level. Further, it uses normalized images for generating the final mosaic. Montage workflow models are generally input-output intensive in comparison to CPU usage.
- SIPHIT workflow: This model category is developed at Harvard University e using a small untranslated RNA. Untranslated RNA allows the controlling of many bacteria processes [21]. This workflow model encodes bacterial replicated geans. Such workflow consists of a task having low processor requirements and high memory requirements for execution.
- Epigenomics workflow: This workflow category is designed by the Pegasus Team and the USC Epigenome centre [23]. They used the concept of bioinformatics automatic genome sequence operation execution. Epigenomics workflow list of task that requires high CPU and low input output operations for their execution.
- LIGO workflow: this workflow model is the abbreviation of laser interferometer gravitational-wave observatory. It is defined based upon detection of gravitational wave network with the observatory in Livingston and Hanford. It uses the concept of Physics like a gravitational wave. LIGO workflow

model comprises task that requires more CPU and large memory during their execution.

- Cyber-Shake workflow: This kind of workflow model is utilized by the Southern California Earth quake Center (SCEC) [24] for characterizing earth quake menaces. It is defined as providing robustness, reliability, and automating the requirement to reach the necessary competition scale. The tasks involved in this workflow model are CPU intensive and memory intensive for their execution.

Several workflow scheduling methods have been validated using the workflow mentioned above, particularly scientific workflows. Scientific workflows are generally generated using many tools developed by different organizations such as Pegasus group, DAGman, Wfms. These tools use real-time execution traces for generating synthetic workflows. Table 1 summarizes different workflow models.

In the cloud computing environment, service providers accept different applications from the end users in workflows. Such applications can run the server either individual workflow or a workflow group. Workflow scheduling methods can be developed to execute workflow as a single instance or for more than one instance, known as workflow multiplicity. Therefore, application models can be e divided into different categories based upon workflow multiplicity as below.

- Single workflow: this category of applications is a single workflow model, and their algorithms are designed for meeting different objectives of the single workflow instance. This model is the conventional model used in the grid environment. This type of model aims to choose optimal and appropriate computing resources for executing different tasks. The workflow tasks are executed in sequence without any dependency on different servers. This workflow model attempts to optimize execution cost and makespan of the overflow under given constraints of quality of service as per service level agreement.
- Multiple workflows: Multiple workflow models are generated by cloud user applications with more than one scientific workflow. Therefore, it requires multiple workflow instances that differ in their properties, structure, computing resource requirements and size. Therefore, workflow scheduling algorithms ensure the deployment of workflow tasks dynamically to the grid computing resources of the cloud server. Multiple workflow models can have different quality of service requirements and are executed without any dependency on the cloud server. It requires developing a workflow scheduling method to find an optimal schedule for each workflow dynamically while meeting quality of service requirements in the cloud data centre. For example, Tolosana-Calasanz et al. [25] developed a dynamic workflow scheduling method that solves multiple workflow problems in the cloud computing environment. They proposed generating multiple pipeline workflow instances, consisting of quality of service requirements for each instance during its processing. This workflow model consists of multiple interdependent tasks, requiring mapping suitable virtual machines in the cloud computing environment.

Table 1. Summary of workflow model types

Workflow model type	**Workflow subtype**
Synthetic workflows	• Sequential workflow • Parallel workflow • State Machine workflow • Rules Driven workflow
Scientific workflows	• Montage workflow • Siphit workflow • Epigenomics workflow • LIGO workflow • Cyber-Shake workflow

- Workflow ensembles: This workflow model consists of different applications generating multiple scientific workflows producing the desired output in combination. Multiple scientific workflows generated in this model are interrelated and executed without dependency on cloud servers. Generally, workflow ensembles consist of workflows with similar structures consisting of different task sizes. They require the mapping of appropriate virtual machines. This workflow model mainly attempts to optimize the makespan of workflows under the quality of service constraints defined in service level agreement. Table 2 presents different application workflow models described above.

Table 2. Application workflow models

Application workflows	• Single workflow • Multiple workflows • Workflow ensembles

3. Workflow Scheduling Types

Several scheduling methods have been developed in the cloud computing environment to effectively schedule different tasks and workflows by optimizing different kinds of objectives [26]. These scheduling methods can be broadly categorized below [27].

- User level scheduling methods: These methods have been developed by focusing on the problems raised during provisioning different services among cloud users and cloud providers.
- System level scheduling methods: this category of scheduling methods attempt to optimize computing resources within the data centre of the cloud computing environment.
- Static scheduling methods: this category of scheduling methods assume simultaneous arrival of tasks, and accordingly, computing resources are scheduled for each task before their execution. These methods assume prior knowledge about task execution requirements such as CPU requirements, deadline constraints etc. This method has low runtime overhead. The most popular static scheduling method is the opportunistic load balancing method.
- Dynamic scheduling methods: these methods do not have prior information about task arrival. Search methods provision computing resources and adapt timing updates dynamically during the execution of the task. It increases the execution

overhead. The most famous example in this category are the earliest deadline first scheduling method.

• Centralized scheduling methods: These methods assume the availability of fe master processor that collects different tasks and transfers them to the other processing elements for further processing as per their requirements [28].

• Distributed scheduling methods: In distributed scheduling, there is a local scheduler for managing the cloud request and keeping a record of different job statuses. Tasks are executed over a distributed network. It has been validated that distributed scheduling is comparatively less efficient as compared to centralized scheduling [29].

• Primitive scheduling methods: these scheduling methods allow executing tasks to get interrupted and migrated or re started later on different machine / computing resources [30].

• Non primitive scheduling methods: This kind of scheduling method does not allow the executing task to get interrupted until its execution finishes. Only on completion of task execution computing resources can be released [31].

• Online scheduling methods: this scheduling method involves scheduling a task only once. It does not allow the change in scheduling results.

• Offline scheduling method: this method do not perform any mapping of computing resources on the arrival of the task. It involves collecting and examining task resource mapping at a given time interval. It is also known as the batch mode heuristic method [32]. Max min and min min scheduling methods are the most popular offline scheduling methods.

• Task level scheduling method: this type of method optimized task to virtual machine allocation in the data centre of a cloud computing environment with the aim of minimizing the running cost of the workflow under given constraints of quality of service [33,34].

• Service level scheduling methods: this category of scheduling methods are mainly concerned with the task to service allocation. It involves mapping workflow tasks as per their quality of service needs.

4. Workflow Scheduling Types

Workflow scheduling methods have been evaluated for its performance in cloud computing environment based upon different performance metrics [35-42].

Several performance metrics have been defined for evaluating workflow scheduling methods. The most popular scheduling objectives include cost and deadline scheduling objectives. Besides, the researchers considered different performance measures such as schedule length ratio, running time, availability of computing servers, budget, reliability, availability, makespan, service level agreement violation, security, load balancing etc. Table 3 presents the different scheduling objectives used to evaluate workflow scheduling methods and is discussed as follows.

• Execution time: workflow execution time indicates the total time required for executing a task in the selected virtual machine. It depends upon task size and computing resources provisioned with the virtual machine. In a workflow, execution time or makespan is the sum of time for executing the task in the workflow. Workflow scheduling algorithms minimize total workflow execution time by setting different deadline constraints.

• Cost: workflow execution cost is determined by virtual machine instances required for executing the task in the workflow, computing resources equipped with virtual machine and task virtual machine mapping method. These methods focus on minimizing total execution cost of the workflow. Workflow scheduling method attempt to find an optimal schedule by assigning the minimum number of virtual machines with appropriate computing resource to different tasks using a suitable task virtual machine mapping method.

• Schedule length ratio: The schedule length ratio gives the ratio of completion time and running time of workflow tasks. The high value of schedule length ratio indicates the task with more completion time. Workflow scheduling methods attempt to find an optimal schedule with a minimum schedule length ratio.

• Computing resource availability: It indicates computing resource availability for executing workflow tasks in parallel [43]. Computing resource availability depends on choosing optimal loaded service and virtual machine deployment strategy. The workflow scheduling method tries to maximize computing resource utilization and computing resource availability of Cloud Service by reducing computing resource wastage in cloud data centres.

• Reliability: it defines the completion of execution of the task in a given interval [43]. It depends upon computing resource capacity and the overall workload of the cloud server. The scheduling algorithm tries to allocate virtual machines to computing resources as per bare resource needs or deadlines.

• Power consumption: Most cloud service providers, organizations, and governments are concerned about reducing power consumption and carbon footprint. Virtualization enabled efficient use of computing resources in a cloud computing environment by provisioning multiple virtual machines over a single physical machine, reducing power consumption of physical machines by a considerable amount. Workflow scheduling algorithms mainly focus on executing workflow appropriate virtual machines to optimize power consumption and minimize carbon footprints from Cloud servers.

• Computing resource utilization: Computing resource utilization defines computing resources' reusability by allocating different virtual machines and workflow tasks. The primary aim of the workflow scheduling method is to optimize computing resource usage for executing workflow tasks quickly.

• Communication overhead: Communication overhead is considered a critical challenge for scheduling methods in cloud computing environments [109] while executing multiple tasks on cloud servers. Scheduling methods mainly focused on reducing communication costs. Communication overhead increases by executing task on multiple physical machines.

• Security: One crucial objective of scheduling methods in the cloud computing environment is security. Specific scientific workflows may require a secure transfer of data [40] during their execution. It may be possible that data get transferred from one machine to another for non-availability of computing resources or other reasons. In such a scenario, scheduling methods must provide a secure way to transfer data from one machine to another to execute the workflow tasks. Table 3 depicts the scheduling objectives mentioned above.

5. Workflow Scheduling Methods

Workflow scheduling methods attempt to allocate computing resources equipped with different virtual machines having different processing capabilities to execute workflow tasks to optimize the given scheduling objectives such as cost, makespan etc. [72]. Several developments have been made in designing efficient workflow scheduling algorithms in the cloud computing environment. For example,

Wu and Lee [56] suggested a scheduling method for minimizing power consumption of workflow in a computing environment consisting of the Internet of things. The authors focused on two kinds of Edge nodes. The first kind of edge nodes is fast nodes with more computational power, requiring more power for its operation. The second type of Edge nodes is slow nodes having low power consumption due to less computational power. They validated their approach experimentally and demonstrated that their algorithm resulted in the best power consumption compared to the conventional longest time first method and random algorithm. However, they have ignored periodic task and task migration time among the nodes.

Table 3. Scheduling objectives

Scheduling objective	Reference
Schedule length ratio	[45] [46]
Execution time	[47]
Availability of computing resources	[48]
Budget	[49]
Reliability	[50]
Makespan	[51]
Service level agreement violation	[52]
Security	[53]
Load balancing	[54]
Resource utilization	[55]

In contrast, Xu et al. [57] proposed using an improved particle swarm optimization algorithm for scheduling workflows in the cloud computing environment. They proposed a new method for updating inertia weights by designing a non linear decreasing function that balances particles' global and local capability in their algorithm. They recorded particles in the particle swarm optimization algorithm as a schedule plan as per the number of tasks in the workflow. Their experimental results demonstrate a considerable reduction in the completion time of workflow in comparison to conventional particle swarm optimization algorithm [58]. However, they have not mentioned about power consumption in their study.

Kabirzadeh et al. [59] introduced a new workflow scheduling algorithm in a fog computing environment using a hyper-heuristic-based method that finds optimal solutions for scheduling workflows. They empirically proved a decrease in power consumption of 69.9% and cost by 59.6 2% in comparison to particle swarm optimization algorithm. This method also optimizes the power consumption and simulation time by assigning computing resources as per condition to execute workflows.

Mao and Humphrey [60] introduced a scheduling algorithm in the cloud computing environment that dynamically schedules workflow by considering different types of virtual machines with different costs based upon workflow requirements. In this study, the authors attempt to address the issues of resource sharing and job scheduling by optimizing Global and job level cost efficiencies. The authors mainly focus on minimizing total execution time using a heuristic approach. Similarly, Malawski et al. [61] proposed one static and two dynamic algorithms for efficient scheduling of workflow tasks in a cloud computing environment by following a mathematical model. Their approach assumes that cloud service providers allocate heterogeneous machines to the cloud users for executing their respective applications within the given time limits. The authors mainly focused on finding the globally optimal solution for scheduling tasks based upon the mixed integer programming.

Abrishami et al. [62] proposed a static workflow scheduling method based upon the partial critical path. They used a heuristic approach for assigning partial critical path tasks to a single virtual machine to minimize the total number of virtual machines located for executing the task. The author attempt to minimize execution time while meeting time deadlines of the workflow in cloud computing environment. It enables minimizing computing resource wastage of Cloud Servers.

Byun et al. [63] presented a cost aware scheduling method for executing workflows dynamically in the cloud computing environment. The authors attempted to minimize the number of cloud servers required to execute workflow as given time limits. They also provided a framework for deploying workflows and providing computing resources to cloud data servers dynamically.

Abrishami et al. [64] introduced a quality of service aware methods for scheduling workflows in a cloud computing environment that find partial critical paths and assign computing resources accordingly. They mainly focused on minimizing the cost of computing resources while meeting the deadline for executing workflow in the cloud computing environment. Their approach used a recursive method to the scheduled workflow task at partial critical parts. This algorithm fulfils the necessary primary conditions of the cloud model, such as flexibility and elasticity.

Ghafarian et al. [66] presented partition based method for scheduling workflow in cloud computing. They Generated sub flows using a partitioning method. They focused on minimizing communication cost and minimizing the execution time of workflow tasks. Their approach involves the distribution of workflow tasks over multiple virtual machines to maintain an appropriate scheduling strategy.

Alkhanak et al. [65] also presented a cost aware method for scheduling workflows in a cloud computing environment by balancing workload among different computing resources. They attempted to optimize computing resource utilization, minimize cost, and make span of workflow execution in the cloud computing environment. They proposed classifying computing resources based on their workflow quality of service parameters.

Artem et al. [67] introduced an effective method for workflow scheduling that minimizes the execution time of the workflows in cloud computing. They mainly focused on minimizing makespan and budget for executing workflow in their approach.

Chirkin et al. [45] developed a workflow scheduling method that dynamically schedules workflows by estimating execution time using an optimal scheduling method called a dual stochastic function. They mainly focused on improving makespan estimation accuracy and increasing algorithm performance in their approach.

Singh et al. [68] also designed a workflow scheduling strategy to achieve cost efficiency and meet deadline conditions dynamically. They applied the mean clustering, and subset sum problem approaches in their workflow scheduling methods.

Lee et al. [69] proposed a method for efficiently utilizing computing resources in executing workflows in the cloud computing environment. They mainly focused on minimizing makespan and maximizing computing resource utilization. They tried to find near optimal solutions for workflow scheduling.

Ye et al. [70] suggested a workflow scheduling method that dynamically schedules workflows and optimizes execution time, makespan reliability and execution cost of workflow execution in the cloud computing environment.

Haidri et al. [71] designed a cost efficient and deadline aware method for scheduling workflows. This method allocates computing resources to the workflow tasks using a rank based method. The proposed method minimizes makespan and cost as per the quality of service conditions defined for the workflows.

Bochenina et al. [72] presented a static method for scheduling workflow in a cloud computing environment that maths multiple workflows to available computing resources. This method chooses the computing resources for executing workflow tasks based on unified metrics that incorporate different levels. It enables meeting deadlines for workflow execution and ensures efficient utilization of computing resources

Yun et al. [73] suggested a two phased scheduling method that dynamically schedules workflows in cloud computing. The first phase days with task scheduling and the second phase is concerned with resource provisioning. Their approach is a generic method that simultaneously considers task mapping and resource provisioning to minimize communication delay.

Arabnejad et al. [75] designed a deadline aware workflow scheduling method in cloud computing. This method allocates appropriate computing resources as per proportional deadline constraints of workflow tasks.

Sahni et al. [74] suggested cost effective method for minimizing workflow execution cost under the given quality of service constraints. This method creates a workflow pipeline and assigns appropriate computing resources to those pipelines. The pipeline contains a collection of interdependent tasks that can be executed in sequence using a given set of computing resources.

Several researchers paid significant attention to metaheuristic algorithms to find an optimal workflow scheduling solution in cloud computing. Metaheuristic algorithms are problem independent technique that use specific heuristics or policies for finding globally optimal solutions. The significant research in metaheuristic based workflow scheduling methods is described as follows. Pandey et al. [76] utilized particle swarm optimization based method for workflow scheduling in cloud computing. The main objective of their work is to optimize the execution cost of workflow tasks while keeping workload balance among available computing resources. They assume that there is a fixed number of virtual machines over heterogeneous servers. The proposed method attempts to find suitable computing resources for executing workflow tasks by optimizing cost and transmission time.

Rodriguez et al. [91] also focused on meta heuristic algorithm for developing a static method of workflow scheduling. They mainly tried to optimize execution cost and makespan for executing workflow tasks while meeting deadline conditions. They assumed that cloud data centre consists of the number of virtual machines with dynamic computing resources that can be provisioned to the cloud users on a lease basis. Using a meta heuristic algorithm called particle Swarm Optimisation, the authors attempted to find optimal computing resources for executing tasks in workflow of a cloud computing environment.

Su et al. [78] introduced Îµ-Fuzzy Dominance sort-based discrete particle swarm optimization method for scheduling workflow in the cloud computing environment. They applied the fuzzy logic concept to find Pareto optimal solutions considering multiple conflicting objectives like exhibition cost and makespan of workflow in the cloud computing environment. This approach assigns workflow tasks to a cost-efficient virtual machine based on Pareto optimal solutions. And it is followed by reducing monetary costs in the cloud computing environment.

Verma et al. [79] also used the hybrid particle swarm optimization method to obtain near optimal solutions by optimizing multiple conflicting objectives of workflow scheduling in cloud computing. They mainly focus on optimizing execution cost and makespan of workflow scheduling.

Barrett et al. [80] applied a genetic algorithm for developing a dynamic workflow task scheduling algorithm. The genetic algorithm obtains an optimal workflow task schedule followed by the Markov decision method to execute workflow tasks.

Jian et al. [81] developed a workflow scheduling algorithm using the simulated annealing method. This method attempted to optimize multiple objectives of workflow scheduling, execution cost and execution time while maintaining a balanced workload over the cloud data centre.

Liu et al. [82] applied a genetic algorithm to propose a workflow scheduling approach. This approach developed an adaptive penalty function for meeting the quality of service conditions of the workflow. The authors suggested a co evolutionary method for adjusting mutation and crossover probability to achieve fast convergence in the cloud computing environment.

Elsherbiny et al. [83] suggested an approach for workflow scheduling based upon an intelligent water drop algorithm. This method finds a near optimal schedule of workflow.

Nasonov et al. [84] proposed a hybrid approach for scheduling workflow in cloud computing using genetic algorithms and heuristic algorithms. A heuristic algorithm is applied for generating an initial population of workflow tasks, followed by applying a genetic algorithm to find the near optimal schedule using mutation and crossover operations.

Shishido et al. [85] proposed a cost aware secure approach for scheduling workflow in a cloud computing environment based upon metaheuristic approaches. They empirically compared the performance of their approach with the genetic algorithm and particle Swarm Optimization algorithm. They concluded that the genetic algorithm performed better in comparison to particle Swarm Optimization algorithm regarding response time and every question cost of workflow in the cloud computing environment.

Choudhary et al. [86] also proposed a hybrid approach to schedule workflow in cloud computing based upon gravitational search algorithm and heuristic approach to find near optimal schedules of workflows. They demonstrated that the hybrid approach is better in comparison to the Heterogeneous Earliest Finish Time algorithm regarding execution cost and makespan.

Wu et al. [87] develop a particle swarm optimization aware algorithm to schedule workflow in the cloud computing environment. This method chooses optimal computing resources based upon the heuristic method for optimizing execution cost and makespan by ensuring computing resource utilization in executing workflow task of cloud computing. The authors assumed a static virtual machine in the cloud computing data centre to execute workflow tasks.

Table 4 summarizes a comparison of above mentioned workflow scheduling methods in cloud computing.

6. Discussion and Future Directions

It can be noted that many researchers have studied performance-based scheduling in different types of systems. However, it can be observed from above cited workflow scheduling methods that there are many heuristic and metaheuristic algorithm-based workflow scheduling methods. Most researchers focus on the particle swarm optimization algorithm due to its fast convergence in finding an optimal workflow schedule in a cloud computing environment. Many researchers focused on hybrid approaches by combining heuristic and metaheuristic algorithms. It can be noticed that hybrid workflow. Scheduling methods perform better than the conventional method due to better population initialization using some heuristic methods.

It can also be observed from the cited workflow scheduling methods that different researchers consider many scheduling objectives [92-94]. Most researchers focused on execution time, exhibition cost, computing resource utilization and deadline. Many authors considered multiple conflicting objectives of workflow scheduling simultaneously using the meta heuristic method in their fitness function. They used particle swarm optimization and genetic algorithm based methods for finding optimal solutions to workflow scheduling problems in cloud computing.
However, many issues have not been studied appropriately in workflow scheduling areas that require the immediate attention of fellow researchers.

- Cost performance programming: It can be noted that performance based scheduling has been widely analyzed in different systems [93]. However, in heterogeneous systems, performance is exchanged by cost performance analysis.

Table 4. Summary of workflow scheduling methods

Study	**Approach**	**Scheduling type**	**Scheduling objective**
Mao et al. [60]	Optimization the global level and job level cost efficiencies	Dynamic scheduling	• Total Execution Cost • Total Execution Time
Malawski et al. [61]	Workflow scheduling using mathematical model and mixed Integer Programming	Dynamic and static scheduling	• Deadline • Total Execution Cost and Time
Abrishami et al. [62]	Partial critical paths (PCP) based method for static workflow scheduling	Static workflow scheduling	• Total Execution Cost and Time Schedule length ratio • Deadline
Byun et al. [63]	Cost aware dynamic workflow scheduling	Dynamic scheduling	• Deadline • Total Execution Cost and Time
Abrishami et al. [64]	QoS-aware workflow scheduling	Dynamic scheduling	• Deadline • Schedule length ratio • Total Execution Time
Alkhanak et al. [65]	Cost aware work flow scheduling	Dynamic scheduling	• Deadline • Total Execution Cost and Time
Ghafarian et al. [66]	Partition based strategy for workflow scheduling	Dynamic scheduling	• Deadline • Total Execution Cost and Time
Chen et al. [67]	Heuristic approach for workflow Scheduling	Dynamic scheduling	• Budget • Schedule length ratio • Total Execution Cost and Time
Chirkin et al. [45]	Dual stochastic functions for workflow Scheduling	Dynamic scheduling	• Total Execution Time • Reliability
Singh et al. [68]	Dynamic provisioning of the resources for workflow scheduling	Dynamic scheduling	• Resource Utilization • Deadline
Lee et al. [69]	Dynamic and efficient computing resource usage for workflow scheduling	Dynamic scheduling	• Resource Utilization • Deadline
Ye et al. [70]	Keen point driven evolutionary method for workflow scheduling	Dynamic scheduling	• Resource Utilization • Total Execution Cost and Time
Haidri et al. [71]	Rank-based policy for workflow scheduling	Dynamic scheduling	• Deadline • Schedule length ratio • Total Execution Cost and Time
Bochenina et al. [72]	Unified metric incorporating levels based method for workflow scheduling	Dynamic scheduling	• Total Execution Time • Resource Utilization • Deadline
Yun et al. [73]	Generic algorithm for workflow scheduling	Dynamic scheduling	• Total Execution Time
Sahni et al. [74]	Pipeline based resource assignment for workflow scheduling	Dynamic scheduling	• Budget • Deadline • Total Execution Cost and Time
Arabnejad et al. [75]	Deadline aware method for workflow scheduling	Dynamic scheduling	• Total Execution Cost • Schedule length ratio • Deadline • Budget
Pandey et al. [76]	Particle Swarm Optimization algorithm for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Wu et al. [91]	Particle Swarm Optimization algorithm for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time

Rodriguez et al. [77]	Particle Swarm Optimization algorithm and heuristic algorithm for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Su et al. [78]	Îμ-Fuzzy Dominance sort-based Discrete Particle Swarm Optimization algorithm for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Verma et al. [79]	Hybrid Particle Swarm Optimization algorithm for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Barrett et al. [80]	Genetic algorithm based method for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Jian et al. [81]	Simulated Annealing algorithm for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Liu et al. [82]	Genetic algorithm based method for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Elsherbiny et al. [83]	intelligent water drops based algorithm based method for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Nasonov et al. [84]	Heuristic algorithms and Genetic algorithm based method for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Shishido et al. [85]	cost-aware genetic algorithm based method for workflow scheduling in cloud computing	Dynamic scheduling	• Budget • Deadline • Reliability • Resource utilization • Schedule length • Total Execution Cost and Time
Choudhary et al. [86]	Genetic algorithm and Particle Swarm Optimization algorithm based method for workflow scheduling in cloud computing	Dynamic scheduling	• Budget, • Deadline • Schedule length • Resource utilization • Reliability • Total Execution Cost and Time

The cloud service provider offers computing resources with different capabilities and prices to meet end user requirements. But it is a challenge for workflow scheduling to analyze cost and performance with heterogeneous computing resources [95]. Nowadays, most commercial cloud computing service providers are charging their end users based upon time for acquiring computing resources, raising new issues for workflow scheduling. The primary issue, in this case, is the computing resource fragment optimization.

- The auto scale of workflow as a service: it is a big challenge for estimating the appropriate amount of computing resources in an integrated scenario of resource provisioning technologies and workflow technologies [96]. Computing resource provisioning strategies minimize economic costs and maximize computing resource usage. However, automatic computing resource scaling to execute workflow focuses on delivering workflow applications as a service [97].
- Robust workflow scheduling: The critical problem in implementing workflow scheduling algorithms in the real environment is the uncertain arrival of workflow tasks for execution and transmission time [98]. Therefore, it becomes necessary to analyze to ensure workflow scheduling performance in cloud computing. Many researchers assume that there is a finite structure of workflows. However, there can be some conditional branches for loops in workflows. It becomes difficult to ensure performance in such cases, requiring significant attention of the research community to guarantee appropriate performance of workflow scheduling in real time scenarios. There can be failures in large cloud data centres due to many reasons, including hardware and software. Failure handling can be a promising direction for future research for robust workflow scheduling in the cloud computing environment [99].
- Data intensive workflow scheduling: most researchers ignored the direct data transfer between workflow tasks [99]. They assume downloading and uploading the data is the path of task execution. But it is not valid in all scenarios. Data intensive applications may involve dominating data based activities compared to execution time. Therefore, it is required to explore

data transfer performance separately and design different placement methods accordingly.

• Integration of multiple environments: Recently, many environments have been integrated, including public and private clouds, leading to the availability of heterogeneous computing resources [100]. It has provided flexibility in choosing computing resources and provides benefits of different computing resources to the end users in cloud computing. However, appropriate mechanisms for power consumption of cloud data centres and resource utilization for the hybrid environment still need to be developed.

• Considering emerging Technologies in workflow scheduling: Several new technologies have been emerged, providing benefits in workflow scheduling methods, like container-based scheduling, serverless computing and fog computing [101]. Containers offer a lightweight framework to execute workflow tasks as a stand alone and self contained unit, enabling customized execution of end user applications in the form of Dockers [88] [89] or Kubernetes [90]. Containers provide several advantages initiating quick Launch for execution, requiring small memory for storing information and executing the applications. Container technology helps improve resource utilization and achieve better performance by executing tasks in parallel.

7. Conclusion

Cloud computing paradigms have been developed to offer a variety of virtual computing resources for executing workflow tasks of end-users effectively and efficiently. Scheduling methods have been developed to map workflow tasks to appropriate virtual machines in the cloud data centre while optimizing different objectives. The most commonly used objectives include power consumption, execution time, execution cost, performance, quality of service requirements etc.

This paper presents workflow scheduling and recent developments in cloud computing environment scheduling workflows. In particular, it provides a taxonomy of scheduling methods, scheduling objectives, and scheduling mechanisms in the cloud computing environment. The paper presents a classification of workflows as synthetic workflows and scientific workflows. Various workflow scheduling strategies have been analyzed and compared based on their approach, scheduling type and scheduling objective. We presented a discussion on trends in workflow scheduling algorithms and presented promising pathways for future research in this area.

REFERENCES

[1]. Adhikari, M., Amgoth, T., & Srirama, S. N. (2019). A survey on scheduling strategies for workflows in cloud environment and emerging trends. ACM Computing Surveys (CSUR), 52(4), 1 - 36.

[2]. Hesham El - Rewini and Ted G. Lewis. 1990. Scheduling parallel program tasks onto arbitrary target machines. J. Parallel Distrib. Comput. 9, 2 (1990), 138 – 153.

[3]. Edwin S. H. Hou, Nirwan Ansari, and Hong Ren. 1994. A genetic algorithm for multiprocessor scheduling. IEEE Trans. Parallel Distrib. Syst. 5, 2 (1994), 113 – 120.

[4]. Yu - Kwong Kwok and Ishfaq Ahmad. 1996. Dynamic critical - path scheduling: An effective technique for allocating task graphs to multiprocessors. IEEE Trans. Parallel Distrib. Syst. 7, 5 (1996), 506 – 521.

[5]. Gilbert C. Sih and Edward A. Lee. 1993. A compile - time scheduling heuristic for interconnection - constrained heterogeneous processor architectures. IEEE Trans. Parallel Distrib. Syst. 4, 2 (1993), 175 – 187.

[6]. Aarti Singh, Dimple Juneja, and Manisha Malhotra. 2017. A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. J. King Saud Univ. Comput. Inf. Sci. 29, 1 (2017), 19 – 28.

[7]. S. Suresh and S. Sakthivel. 2017. A novel performance constrained power management framework for cloud computing using an adaptive node scaling approach. Comput. Electr. Eng. 60 (2017), 30 – 44.

[8]. Chatterjee, T., Ojha, V. K., Adhikari, M., Banerjee, S., Biswas, U., & Snášel, V. (2014). Design and implementation of an improved datacenter broker policy to improve the QoS of a cloud. In Proceedings of the Fifth International Conference on Innovations in Bio - Inspired Computing and Applications IBICA 2014 (pp. 281 - 290). Springer, Cham.

[9]. Google App Engine. Retrieved from https://cloud.google.com/appengine.

[10]. Salesforce.com. Retrieved from http://www.salesforce.com/au/saas.

[11]. Microsoft Azure. Retrieved from https://azure.microsoft.com.

[12]. Elastic Compute Cloud EC2. Retrieved from http://aws.amazon.com/ec2.

[13]. Wu, F., Wu, Q., & Tan, Y. (2015). Workflow scheduling in cloud: a survey. The Journal of Supercomputing, 71(9), 3373 - 3418.

[14]. GideonJuve and Ewa Deelman.2011. Scientific workflows in the cloud. In Grids, Clouds and Virtualization. Springer, Berlin, 71 – 91.

[15]. Marek Wieczorek, Radu Prodan, and Thomas Fahringer. 2005. Scheduling of scientific workflows in the ASKALON grid environment. ACM SIGMOD Rec. 34, 3 (2005), 56 – 62.

[16]. Duncan A. Brown, Patrick R. Brady, Alexander Dietz, Junwei Cao, Ben Johnson, and John McNabb. 2007. A case study on the use of workflow technologies for scientific analysis: Gravitational wave data analysis. In Workflows for e - Science. Springer, Berlin, 39 – 59.

[17]. Angela O'Brien, Steven Newhouse, and John Darlington. 2004. Mapping of scientific workflow within the e - protein project to distributed resources. In Proceedings of the UK e - Science All Hands Meeting. 404 – 409.

[18]. Radu Prodan and Thomas Fahringer. 2005. Dynamic scheduling of scientific workflow applications on the grid: A case study. In Proceedings of the 2005 ACM Symposium on Applied Computing. ACM, 687 – 694.

[19]. Zhiao Shi and Jack J. Dongarra. 2006. Scheduling workflow applications on processors with different capabilities. Fut. Gen. Comp. Syst. 22, 6 (2006), 665 – 675.

[20]. G. B. Berriman, A. C. Laity, J. C. Good, J. C. Jacob, D. S. Katz, E. Deelman, G. Singh, M. H. Su, and T. A. Prince. 2006. Montage: The architecture and scientific applications of a national virtual observatory service for computing astronomical image mosaics. In Proceedings of the Earth Sciences Technology Conference.

[21]. Jonathan Livny, Hidayat Teonadi, Miron Livny, and Matthew K. Waldor. 2008. High - throughput, kingdom - wide prediction and annotation of bacterial non - coding RNAs. PloS One 3, 9 (2008), e3197.

[22]. G. Bruce Berriman, Ewa Deelman, John C. Good, Joseph C. Jacob, Daniel S. Katz, Carl Kesselman, Anastasia C. Laity, Thomas A. Prince, Gurmeet Singh, and Mei - Hu Su. 2004. Montage: A grid - enabled engine for delivering custom science - grade mosaics on demand. In Optimizing Scientific Return for Astronomy through Information Technologies, Vol. 5493. International Society for Optics and Photonics, 221 – 233.

[23]. USC Epigenome Center. Retrieved from http://epigenome.usc.edu.

[24]. Robert Graves, Thomas H. Jordan, Scott Callaghan, Ewa Deelman, Edward Field, Gideon Juve, Carl Kesselman, Philip Maechling, Gaurang Mehta, Kevin Milner, et al. 2011. CyberShake: A physics - based seismic hazard model for southern California. Pure Appl. Geophys. 168, 3 – 4 (2011), 367 – 381.

[25]. Rafael Tolosana - Calasanz, José Ángel Bañares, Congduc Pham, and Omer F. Rana. 2012. Enforcing qos in scientific workflow systems enacted over cloud infrastructures. J. Comput. Syst. Sci. 78, 5 (2012), 1300 – 1315.

[26]. Masdari, M., ValiKardan, S., Shahi, Z., & Azar, S. I. (2016). Towards workflow scheduling in cloud computing: a comprehensive analysis. Journal of Network and Computer Applications, 66, 64 - 82.

[27]. Chawla Y, Bhonsle M. A study on scheduling methods in cloud computing. Int J Emerg Trends Technol Comput Sci 2012 ; 1(3):12 – 7.

[28]. Kaleeswaran A, Ramasamy V, Vivekanandan P. Dynamic scheduling of data using genetic algorithm in cloud computing. Park Coll Eng Technol 1963.

[29]. Tiwari SM., Cloud computing using cloud - level scheduling: a survey.

[30]. Patel S, Bhoi U. Priority based job scheduling techniques in cloud computing: a systematic review. Int J Sci Techno Res 2013 ; 2:147 – 52.

[31]. Xhafa F, Abraham A. Computational models and heuristic methods for Grid scheduling problems. Future Gener Comput Syst 2010 ; 26(4):608 – 21.

[32]. Vijayalakshmi R, Vasudevan V. Static batch mode heuristic algorithm for mapping independent tasks in computational grid. J Comput Sci 2015 ; 11(1):224.

[33]. Wu Z, et al. A market - oriented hierarchical scheduling strategy in cloud workflow systems. J Super comput 2013 ; 63(1):256 – 93.

[34]. Hong I, Potkonjak M. Power optimization in disk - based real - time application specific systems. In: Proceedings of the 1996 IEEE/ACM international conference on Computer - aided design ; 1997. IEEE Computer Society

[35]. Saeid Abrishami and Mahmoud Naghibzadeh. 2012. Deadline - constrained workflow scheduling in software as a service cloud. Sci. Iran. 19, 3 (2012), 680 – 689.

[36]. Saima Gulzar Ahmad, Chee Sun Liew, Ehsan Ullah Munir, Tan Fong Ang, and Samee U. Khan. 2016. A hybrid genetic algorithm for optimization of scheduling workflow applications in heterogeneous computing systems. J. Parallel Distrib. Comput. 87 (2016), 80 – 90.

[37]. Enda Barrett, Enda Howley, and Jim Duggan. 2011. A learning architecture for scheduling workflow applications in the cloud. In Proceedings of the 9th IEEE European Conference on Web Services (ECOWS'11). IEEE, 83 – 90.

[38]. Eun - Kyu Byun, Yang - Suk Kee, Jin - Soo Kim, and Seungryoul Maeng. 2011. Cost optimized provisioning of elastic resources for application workflows. Future Gen. Comp. Syst. 27, 8 (2011), 1011 – 1026.

[39]. 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 Generation Computer Systems, 74, 1 - 11.

[40]. Artem M. Chirkin, Adam S. Z. Belloum, Sergey V. Kovalchuk, Marc X. Makkes, Mikhail A. Melnik, Alexander A. Visheratin, and Denis A. Nasonov. 2017. Execution time estimation for workflow scheduling. Fut. Gen. Comp. Syst. 75 (2017), 376 – 387.

[41]. Coutinho, R. D. C., Drummond, L. M., Frota, Y., & de Oliveira, D. (2015). Optimizing virtual machine allocation for parallel scientific workflows in federated clouds. Future Generation Computer Systems, 46, 51 - 68.

[42]. Shaymaa Elsherbiny, Eman Eldaydamony, Mohammed Alrahmawy, and Alaa Eldin Reyad. 2018. An extended intelligent water drops algorithm for workflow scheduling in cloud computing environment. Egypt. Inf. J. 19, 1 (2018), 33 – 55.

[43]. Gill, S. S., Buyya, R., Chana, I., Singh, M., & Abraham, A. (2018). BULLET: particle swarm optimization based scheduling technique for provisioned cloud resources. Journal of Network and Systems Management, 26(2), 361 - 400.

[44]. Zhangjun Wu, Zhiwei Ni, Lichuan Gu, and Xiao Liu. 2010. A revised discrete particle swarm optimization for cloud workflow scheduling. In Proceedings of the International Conference on Computational Intelligence and Security (CIS'10). IEEE, 184 – 188.

[45]. Vishakha Singh, Indrajeet Gupta, and Prasanta K. Jana. 2018. A novel cost - efficient approach for deadline - constrained workflow scheduling by dynamic provisioning of resources. Fut. Gen. Comp. Syst. 79 (2018), 95 – 110.

[46]. N. Sadhasivam and P. Thangaraj. 2017. Design of an improved PSO algorithm for workflow scheduling in cloud computing environment. Intell. Autom. Soft Comput. 23, 3 (2017), 493 – 500.

[47]. Rodriguez MA, Buyya R. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. Cloud Comput IEEE Trans 2014 ; 2 (2):222 – 35.

[48]. Sajid M, Raza Z. Cloud computing: issues and challenges. In: Proceedings of the international conference on cloud, big data and trust ; 2013.

[49]. Mary NABaKJ. An extensive survey on QoS in cloud computing. Int J Comput Sci Inform Technol 2014 ; 5:1.

[50]. Zhao L, Ren Y, Sakurai K. Reliable workflow scheduling with less resource redundancy. Parallel Comput 2013 ; 39(10):567 – 85.

[51]. Lopez MM, Heymann E, Senar M. Analysis of dynamic heuristics for workflow scheduling on grid systems. In: Proceedings of the fifth international symposium on parallel and distributed computing, 2006. ISPDC'06 ; 2006. IEEE.

[52]. Alkhanak EN, Lee SP, Khan SUR. Cost - aware challenges for workflow scheduling approaches in cloud computing environments: taxonomy and opportunities. Future Gener Comput Syst 2015.

[53]. Tao Q, et al. QoS constrained grid workflow scheduling optimization based on a novel PSO algorithm. In: Proceedings of the eighth international conference on grid and cooperative computing, 2009. GCC'09 ; 2009. IEEE.

[54]. Anju Baby J. A survey on honey bee inspired load balancing of tasks in cloud computing. In: Proceedings of the international journal of engineering research and technology ; 2013. ESRSA Publications.

[55]. Satish Narayana Srirama, Oleg Batrashev, Pelle Jakovits, and Eero Vainikko. 2011. Scalability of parallel scientific applications on the cloud. Sci. Program. 19, 2 – 3 (2011), 91 – 105.

[56]. H.Y. Wu, C.R. Lee, Energy efficient scheduling for heterogeneous fog computing architectures, Proceedings of the 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), IEEE, Tokyo, Japan, 2018, pp. 555 – 560.

[57]. R. Xu, Y. Wang, Y. Cheng, Y. Zhu, Y. Xie, A.S. Sani, et al., Improved particle swarm optimization based workflow scheduling in cloud - fog environment, International Conference on Business Process Management, Springer, Cham, 2018, pp. 337 – 347.

[58]. J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of ICNN'95 - International Conference on Neural Networks, IEEE, Perth, WA, Australia, 1995, pp. 1942 – 1948.

[59]. S. Kabirzadeh, D. Rahbari, M. Nickray, A hyper heuristic algorithm for scheduling of fog networks, Proceedings of the 2017 21st Conference of Open Innovations Association (FRUCT), IEEE, Helsinki, Finland, 2017, pp. 148 – 155.

[60]. Ming Mao and Marty Humphrey. 2011. Auto - scaling to minimize cost and meet application deadlines in cloud workflows. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC'11). IEEE, 1 – 12.

[61]. M. Malawski, G. Juve, E. Deelman, and J. Nabrzyski. 2012. Cost - and deadline - constrained provisioning for scientific workflow ensembles in iaas clouds. In Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis, 22.

[62]. Eun - Kyu Byun, Yang - Suk Kee, Jin - Soo Kim, and Seungryoul Maeng. 2011. Cost optimized provisioning of elastic resources for application workflows. Future Gen. Comp. Syst. 27, 8 (2011), 1011 – 1026.

[63]. Saeid Abrishami and Mahmoud Naghibzadeh. 2012. Deadline - constrained workflow scheduling in software as a service cloud. Sci. Iran. 19, 3 (2012), 680 – 689.

[64]. Ehab Nabiel Alkhanak, Sai Peck Lee, and Saif Ur Rehman Khan. 2015. Cost - aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. Fut. Gen. Comp. Syst. 50 (2015), 3 – 21.

[65]. Toktam Ghafarian and Bahman Javadi. 2015. Cloud - aware data intensive workflow scheduling on volunteer computing systems. Fut. Gen. Comp. Syst. 51 (2015), 87 – 97.

[66]. Chen, Weihong, Guoqi Xie, Renfa Li, Yang Bai, Chunnian Fan, and Keqin Li. "Efficient task scheduling for budget constrained parallel applications on heterogeneous cloud computing systems." Future Generation Computer Systems 74 (2017): 1 - 11.

[67]. Artem M. Chirkin, Adam S. Z. Belloum, Sergey V. Kovalchuk, Marc X. Makkes, Mikhail A. Melnik, Alexander A. Visheratin, and Denis A. Nasonov. 2017. Execution time estimation for workflow scheduling. Fut. Gen. Comp. Syst. 75 (2017), 376 – 387.

[68]. Aarti Singh, Dimple Juneja, and Manisha Malhotra. 2017. A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. J. King Saud Univ. Comput. Inf. Sci. 29, 1 (2017), 19 – 28.

[69]. Young Choon Lee, Hyuck Han, and Albert Y. Zomaya. 2014. On resource efficiency of workflow schedules. Proc. Comput. Sci. 29 (2014), 534 – 545.

[70]. Xin Ye, Sihao Liu, Yanli Yin, and Yaochu Jin. 2017. User - oriented many - objective cloud workflow scheduling based on an improved knee point driven evolutionary algorithm. Knowl. - Based Syst. 135 (2017), 113 – 124.

[71]. Raza Abbas Haidri, Chittaranjan Padmanabh Katti, and Prem Chandra Saxena. 2017. Cost effective deadline aware scheduling strategy for workflow applications on virtual machines in cloud computing. J. King Saud Univ. Comput. Inf. Sci. (2017). DOI : https://doi.org/10.1016/j.jksuci.2017.10.009

[72]. Bochenina, K., Butakov, N., & Boukhanovsky, A. (2016). Static scheduling of multiple workflows with soft deadlines in non - dedicated heterogeneous environments. Future Generation Computer Systems, 55, 51 - 61.

[73]. Daqing Yun, Chase Qishi Wu, and Yi Gu. 2015. An integrated approach to workflow mapping and task scheduling for delay minimization in distributed environments. J. Parallel Distrib. Comput. 84 (2015), 51 – 64.

[74]. Jyoti Sahni and Deo Vidyarthi. 2018. A cost - effective deadline - constrained dynamic scheduling algorithm for scientific workflows in a cloud environment. IEEE Trans. Cloud Comput. 6, 1 (2018), 2 – 18. DOI:10.1109/JIOT.2018.2838022

[75]. Vahid Arabnejad, Kris Bubendorfer, and Bryan Ng. 2017. Scheduling deadline constrained scientific workflows on dynamically provisioned cloud resources. Fut. Gen. Comp. Syst. 75 (2017), 348 – 364.

[76]. Suraj Pandey, Linlin Wu, Siddeswara Mayura Guru, and Rajkumar Buyya. 2010. A particle swarm optimization based heuristic for scheduling workflow applications in cloud computing environments. In Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'10). IEEE, Los Alamitos, CA, 400 – 407.

[77]. Maria Alejandra Rodriguez and Rajkumar Buyya. 2014. Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. IEEE Trans. Cloud Comput. 2, 2 (2014), 222 – 235.

[78]. Sen Su, Jian Li, Qingjia Huang, Xiao Huang, Kai Shuang, and Jie Wang. 2013. Cost - efficient task scheduling for executing large programs in the cloud. Parallel Comput. 39, 4 – 5 (2013), 177 – 188.

[79]. Amandeep Verma and Sakshi Kaushal. 2017. A hybrid multi - objective particle swarm optimization for scientific workflow scheduling. Parallel Comput. 62 (2017), 1 – 19.

[80]. Enda Barrett, Enda Howley, and Jim Duggan. 2011. A learning architecture for scheduling workflow applications in the cloud. In Proceedings of the 9th IEEE European Conference on Web Services (ECOWS'11). IEEE, 83 – 90.

[81]. Chengfeng Jian, Yekun Wang, Meng Tao, and Meiyu Zhang. 2013. Time - constrained workflow scheduling in cloud environment using simulation annealing algorithm. J. Eng. Sci. Technol. Rev. 6, 5 (2013).

[82]. Li Liu, Miao Zhang, Rajkumar Buyya, and Qi Fan. 2017. Deadline - constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing. Concurr. Comput.: Pract. Exper. 29, 5 (2017), e3942.

[83]. Shaymaa Elsherbiny, Eman Eldaydamony, Mohammed Alrahmawy, and Alaa Eldin Reyad. 2018. An extended intelligent water drops algorithm for workflow scheduling in cloud computing environment. Egypt. Inf. J. 19, 1 (2018), 33 – 55.

[84]. Nasonov, D., Butakov, N., Balakhontseva, M., Knyazkov, K., & Boukhanovsky, A. V. (2014). Hybrid evolutionary workflow scheduling algorithm for dynamic heterogeneous distributed computational environment. In International Joint Conference SOCO'14 - CISIS'14 - ICEUTE'14 (pp. 83 - 92). Springer, Cham.

[85]. Henrique Yoshikazu Shishido, Júlio Cezar Estrella, Claudio Fabiano Motta Toledo, and Marcio Silva Arantes. 2017. Genetic - based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds. Computers & Electrical Engineering 69 (2018), 378 – 394.

[86]. Anubhav Choudhary, Indrajeet Gupta, Vishakha Singh, and Prasanta K. Jana. 2018. A GSA - based hybrid algorithm for bi - objective workflow scheduling in cloud computing. Fut. Gen. Comp. Syst. 83 (2018), 14 – 26.

[87]. Zhangjun Wu, Zhiwei Ni, Lichuan Gu, and Xiao Liu. 2010. A revised discrete particle swarm optimization for cloud workflow scheduling. In Proceedings of the International Conference on Computational Intelligence and Security (CIS'10). IEEE, 184 – 188.

[88]. Alfonso Pérez, Germán Moltó, Miguel Caballer, and Amanda Calatrava. 2018. Serverless computing for container based architectures. Fut. Gen. Comp. Syst. 83 (2018), 50 – 59.

[89]. Dirk Merkel. 2014. Docker: Lightweight linux containers for consistent development and deployment. Linux J. 2014, 239 (2014), 2.

[90]. Víctor Medel, Rafael Tolosana - Calasanz, José Ángel Bañares, Unai Arronategui, and Omer F. Rana. 2018. Characterising resource management performance in Kubernetes. Comput. Electr. Eng. 68 (2018), 286 – 297.

[91]. Zhangjun Wu, Zhiwei Ni, Lichuan Gu, and Xiao Liu. 2010. A revised discrete particle swarm optimization for cloud workflow scheduling. In Proceedings of the International Conference on Computational Intelligence and Security (CIS'10). IEEE, 184 – 188.

[92]. Iranmanesh, A., & Naji, H. R. (2021). DCHG - TS: a deadline - constrained and cost - effective hybrid genetic algorithm for scientific workflow scheduling in cloud computing. Cluster Computing, 24 (2), 667 - 681.

[93]. Mohammadzadeh, A., Masdari, M., & Gharehchopogh, F. S. (2021). Energy and cost - aware workflow scheduling in cloud computing data centers using a multi - objective optimization algorithm. Journal of Network and Systems Management, 29 (3), 1 - 34.

[94]. Doostali, S., Babamir, S. M., & Eini, M. (2021). CP - PGWO: multi - objective workflow scheduling for cloud computing using critical path. Cluster Computing, 24(4), 3607 - 3627.

[95]. Belgacem, A., & Beghdad - Bey, K. (2021). Multi - objective workflow scheduling in cloud computing: trade - off between makespan and cost. Cluster Computing, 1 - 17.

[96]. Bacanin, N., Zivkovic, M., Bezdan, T., Venkatachalam, K., & Abouhawwash, M. (2022). Modified firefly algorithm for workflow scheduling in cloud - edge environment. Neural Computing and Applications, 1 - 26.

[97]. Lei, J., Wu, Q., & Xu, J. (2022). Privacy and security - aware workflow scheduling in a hybrid cloud. Future Generation Computer Systems.

[98]. Chakravarthi, K. K., Neelakantan, P., Shyamala, L., & Vaidehi, V. (2022). Reliable budget aware workflow scheduling strategy on multi - cloud environment. Cluster Computing, 1 - 17.

[99]. Soma, P., Latha, B., & Vijaykumar, V. (2022). An Improved Multi - Objective Workflow Scheduling Using F - NSPSO with Fuzzy Rules. Wireless Personal Communications, 1 - 23.

[100]. Mboula, J. E., Kamla, V. C., Hilman, M. H., & Djamegni, C. T. (2022). Energy - efficient workflow scheduling based on workflow structures under deadline and budget constraints in the cloud. arXiv preprint arXiv : 2201.05429.

[101]. Bisht, J., & Vampugani, V. S. (2022). Load and Cost - Aware Min - Min Workflow Scheduling Algorithm for Heterogeneous Resources in Fog, Cloud, and Edge Scenarios. International Journal of Cloud Applications and Computing (IJCAC), 12 (1), 1 - 20.