symmetry

MDPI

Review

A Survey on QoS Requirements Based on Particle Swarm Optimization Scheduling Techniques for Workflow Scheduling in Cloud Computing

Mazen Farid [1,2,*], Rohaya Latip [1,3,*], Masnida Hussin [1] and Nor Asilah Wati Abdul Hamid [1]

[1] Department of Communication Technology and Networks, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia; masnida@upm.edu.my (M.H.); asila@upm.edu.my (N.A.W.A.H.)

[2] Faculty of Education-Saber, University of Aden, Aden 2408, Yemen

[3] Institute for Mathematical Research (INSPEM), Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia

* Correspondence: mazenfareed7@yahoo.com (M.F.); rohayalt@upm.edu.my (R.L.)

Received: 23 January 2020; Accepted: 15 February 2020; Published: 4 April 2020

check for updates

Abstract: Cloud computing is an innovative technology that deploys networks of servers, located in wide remote areas, for performing operations on a large amount of data. In cloud computing, a workflow model is used to represent different scientific and web applications. One of the main issues in this context is scheduling large workflows of tasks with scientific standards on the heterogeneous cloud environment. Other issues are particular to public cloud computing. These include the need for the user to be satisfied with the quality of service (QoS) parameters, such as scalability and reliability, as well as maximize the end-users resource utilization rate. This paper surveys scheduling algorithms based on particle swarm optimization (PSO). This is aimed at assisting users to decide on the most suitable QoS consideration for large workflows in infrastructure as a service (IaaS) cloud applications and mapping tasks to resources. Besides, the scheduling schemes are categorized according to the variant of the PSO algorithm implemented. Their objectives, characteristics, limitations and testing tools have also been highlighted. Finally, further directions for future research are identified.

Keywords: cloud computing; cost; IaaS; Makespan; particle swarm optimization; quality of service; reliability; workflow scheduling

1. Introduction

Cloud computing is a new technology that provides virtual, scalable and dynamic resources to users based on a pay-as-you-use service. This technology is network-dependent. Due to the network scale involved, some services such as e-commerce applications use up the entire network [1]. Cloud computing is practically developed in three stages: distributed computing, parallel computing and grid computing [2]. Cloud computing is also used for executing scientific workflows such as geological, astronomical, biological, cosmic, biotechnology and image processing. Cloud computing can be classified into three types: *public clouds* that everyone can register and use for their services, *private clouds* that are operated without limitation of network bandwidth, security visibility and regulatory specifications within the organization and *hybrid clouds* that merge private clouds with public cloud resources. An important aspect of cloud computing is workflow scheduling which is the process of mapping dependent tasks to the available resources considering the quality of service (QoS) constraints [3].

Workflow scheduling incurs a large communication and computation cost [4]. Some meta-heuristic-based algorithms such as ant colony optimization (ACO), genetic algorithm (GA), simulated annealing (SA) and particle swarm optimization (PSO) have been proposed for scheduling tasks or workflows in the cloud environment [5]. As one of the meta-heuristic approaches used to solve optimization problems, PSO initially provides a random solution population then looks for optimal solutions by updating generations. Unlike other meta-heuristic algorithms, there are no developmental operators such as crossover and mutation in the traditional PSO algorithm [6]. Possible solutions in PSO (called particles) travel the problem space by tracking the existing optimal particles [7].

One of the fundamental challenges of cloud computing is how to manage the QoS in terms of reliability and performance during the workflow scheduling process. This involves scheduling complex workflows to concurrently reduce execution time and cost using heterogeneous resources of the cloud [8]. Most of the previous studies focus on only one objective, such as reducing the execution time or minimizing the total execution cost, to optimize the performance of workflow scheduling applications while satisfying user's QoS constraints. However, the complex nature of dynamic workflow necessitates that total execution cost is traded-off for processing time to help strike a balance between them.

1.1. Motivation

The above-mentioned challenge motivates this systematic review that focuses on the QoS requirements of PSO-based algorithms built for scheduling workflows in the cloud environment. The schemes are classified according to the type of PSO algorithm used. Their objectives and properties are also highlighted. In addition, we show the enhancements made by hybridizing PSO algorithms with other meta-heuristic algorithms. This analysis compares the schemes and discusses the peculiarities of each scheme to motivate further research in this field.

1.2. Related Works

Most of the previous surveys on PSO focused on task and workflow scheduling in cloud computing without considering future direction and open issues and some quality of service metrics. Some surveys [9,10] have also reviewed task scheduling algorithms based on PSO. However, their works fall out of the scope of this paper. They mentioned the basic working principles of several scheduling algorithms but did not discuss the pros and cons of various approaches. In contrast to the previous surveys, we systematically review the literature on PSO-based workflow scheduling algorithms in cloud computing while taking account of QoS metrics such as fault tolerance, execution time and cost. Moreover, future directions are highlighted. Table 1 compares this paper with the previous reviews on workflow scheduling using PSO-based algorithms.

1.3. Contributions

The contributions of this survey are as follows:

- Classification of PSO-based workflow scheduling algorithms in cloud computing. The QoS constraints, type of workflow used, advantages and disadvantage of these algorithms in cloud computing are indicated.
- Identification of various quality of service metrics used in the literature. Table 2 defines most of these QoS metrics.
- Identification of bi-objective, tri-objective and multi-objective PSO-based scheduling approaches in the literature.
- Presentation of future directions on state-of-the-art PSO-based scheduling algorithms.

The rest of this paper is organized as follows: Section 2 illustrates the systematic review process adopted to select the relevant research articles for our classification, while Section 3 provides the background to this study. Section 4 discusses the PSO-based scheduling strategies in cloud computing. Section 5 classifies the PSO-based scheduling schemes. Section 6 summarizes the literature review indicating the limitations of the review. Section 7 provides a technical comparison of Cloud, Fog and edge. Open challenges and future research directions are presented in Section 8. Finally, Section 9 concludes this paper.

Table 1. Comparison of previous reviews on workflow scheduling based on PSO.

Year [ref]	Article Title	Quality of Service (QoS)	Testing Tools	Workflow Scheduling	Task Scheduling	Fault Tolerance	Energy Consumption	PSO Algorithms Classification	Future Directions	PSO Only
2013 [9]	Review Paper on PSO in workflow scheduling and Cloud Model enhancing Search mechanism in Cloud Computing	✓	✓	✓	✗	✗	✗	✗	✗	✓
2016 [10]	A Survey of PSO-Based Scheduling Algorithms in Cloud Computing	✓	✓	✓	✓	✗	✓	✓	✗	✓
2016 [11]	Towards workflow scheduling in cloud computing: A comprehensive analysis	✓	✗	✓	✗	✗	✗	✗	✓	✗
2019 [12]	A Survey on Scheduling Strategies for Workflows in Cloud Environment and Emerging Trends	✓	✗	✓	✗	✗	✓	✗	✓	✗
2019 [6]	A comprehensive survey for scheduling techniques in cloud computing	✓	✗	✓	✓	✓	✓	✓	✓	✗
2019 [13]	Load Balancing and Server Consolidation in Cloud Computing Environments: A Meta-Study	✓	✗	✗	✓	✗	✓	✗	✓	✗
This paper	A Survey on QoS requirement based on Particle Swarm Optimization Scheduling Techniques for Workflow Scheduling in Cloud Computing	✓	✓	✓	✗	✓	✓	✓	✓	✓

2. Systematic Review Process

This section gives a clear description of how this review is defined, analyzed and interpreted. Research questions are developed to clearly outline the goals of this study. A search protocol is then designed so that the most relevant research papers can be reviewed. This we achieved by using search strings as well as selected digital libraries. The inclusion and exclusion criteria are defined to determine the parameters and research articles that will be included or excluded during the review process. Thereafter, the data obtained are synthesized to ensure all study questions are answered meaningfully.

2.1. Research Question (RQ)

The research questions below were formulated in the planning stage of this review and their answers are provided in subsequent sections:

RQ1. Which heuristic, meta-heuristic or hybrid PSO technologies are available to support workflow scheduling?
RQ2. Which simulation tool is mostly used to conduct cloud computing experiments?
RQ3. What are the flaws of the current PSO-based workflow scheduling strategies?
RQ4. Which PSO-based workflow scheduling algorithm performs best for different QoS constraints?
RQ5. What are the prospects for PSO-based workflow scheduling schemes?

To answer these questions, we follow the Systematic Literature Review (SLR) protocols [14].

2.2. Search Strategy

For search string construction, the following steps were followed [15]: (1) Selecting the keywords from the RQs. (2) Discovering the synonyms and alternate spelling systems to extend the keywords. (3) Investigating the resulting keywords in the relevant literature. (4) Combining the synonyms and alternate spelling using 'Boolean OR' (5) Combining the main terms and conditions using 'Boolean AND'. Thereafter, we found the general terms and conditions relating to QoS customization. The corresponding search string used in the digital library engines is: "((“multi objective” OR “multi-objective”) AND (“Particle Swarm Optimization” OR “PSO”) AND (“Workflow Scheduling”) AND (“Cloud Computing” OR “Cloud”) AND (“QoS”))".

2.3. Quest Approach (QA)

The research methodology of this analysis has been designed to ensure specific and unprejudiced solutions in the literature are considered for providing answers to the RQs. The defined search parameters are cloud computing, optimization, workflows, scheduling, heuristic techniques, meta-heuristic, and hybrid optimization.

2.4. Sample Discrimination Strategy (SDS)

Initially, 474 research papers were collected and screened for the review using the criteria specified. Some research papers were withdrawn because their titles did not fit into the scope of our current survey and/or they have incomplete descriptions and conclusions. 150 research papers were read but the contents of these articles are not relevant to our subject; hence, they were excluded. After the analysis, we got 79 relevant research papers for our survey of PSO-based workflow scheduling. These papers can be found in the list of references. The research papers were published within the period from 2000 to 2019 with some important historical references [16,17].

2.5. Data Clarification and Planning (DCP)

To organize and compare the required information, extensive structures (Tables 3–8) were generated. Then this information was further arranged to respond to the targeted RQs.

2.6. Target Audience

The survey is targeted at researchers interested in PSO algorithm for scheduling workflows based on specific QoS constraints. Scientists with a broad range of backgrounds including distributed computing, cloud, Big Data, Grid and parallel computing can also benefit from learning how workflows can be scheduled using PSO-based techniques that have been influenced by research over the past few decades. PSO algorithm has found broad use in several fields of computer science and applied mathematical applications including neural network weight calculation, time series analysis, market optimization and many more.

2.7. Sources of Data

Electronic databases, including IEEE Xplore, ResearchGate, Google Scholar, Wiley Library, etc. were searched for research papers on scheduling strategies based on PSO. The following online electronic databases were used for this research:

(1) IEEE Xplore (<www.ieeexplore.ieee.org>).
(2) Science Direct (<www.scidirect.com>).
(3) Springer (<www.springer.com>).
(4) ResearchGate (<www.researchgate.net>).
(5) Google Scholar (<www.scholar.google.co.in>).
(6) Scopus (<www.scopus.com>).
(7) Taylor & Francis (<taylorandfrancis.com>).
(8) Wiley Library (<www.onlinelibrary.wiley.com>).

3. Background

3.1. Workflow

A workflow is a sequence of activities carried out to achieve a defined objective in any environment. It is a group of simple processes that are used for solving complex problems [18]. These processes follow a certain order to improve the execution procedure and ensure efficiency. Workflows define the way various tasks are configured, performed and tracked. It can be modelled as a Direct Acyclic Graph (DAG) consisting of nodes and edges (Figure 1). Workflows can be represented as $W = (T, E)$, where T is a set of tasks $t_1, t_2, \ldots, t_n$ and E is an edge (t_a, t_b) [19].

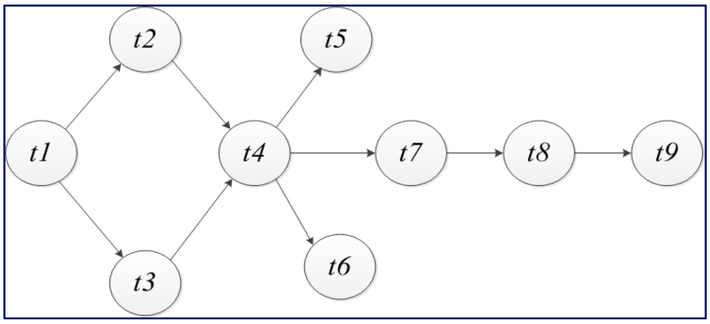


Figure 1. A simple DAG workflow.

A schedule is denoted as:

$$Sched = (Res,\ map,\ CE, TE) \tag{1}$$

where $Res = r_1, r_2, \ldots, r_n$ represents the resources, *map* is the mapping of task-to-resource, CE is the total execution cost and TE is the total execution time [19].

3.2. Scientific Workflow

Several workflows with scientific standards are used in scheduling processes to allocate tasks to appropriate resources. They are also used to measure the efficiency of scheduling methods in different scientific areas. Each scientific workflow contains tasks arranged in levels in the form of a parent-child relationship. Based on this relationship, a parent must be processed first before the child. Several workflow scheduling methods work with different scientific workflow datasets considering that some datasets are complicated and huge. These datasets are processed to meet the users' requirements without violating the considered constraints. Scientific-oriented workflows cover a wide range of areas such as geology, astronomy, biology, cosmic analysis, biotechnology as well as image processing. Some of the most popular workflows are highlighted below and illustrated in Figure 2:

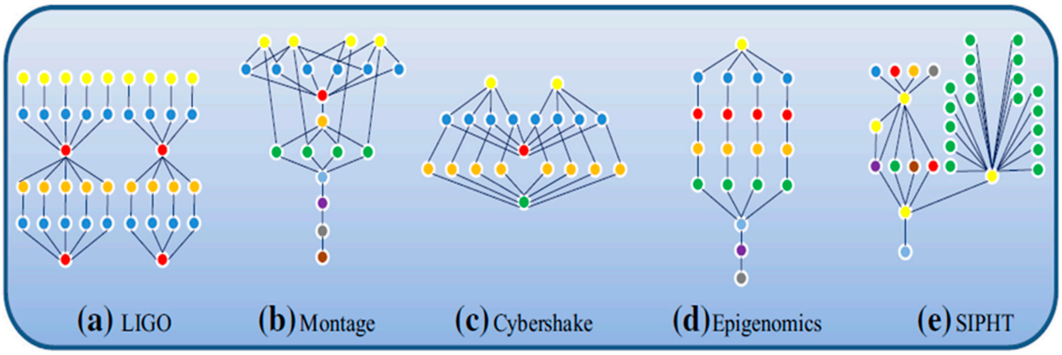


Figure 2. Examples of five well-known scientific workflows from different research areas [18].

- GENOMICS: This workflow is used for DNA methylation and histone data modification in Epigenome Centers (https://pegasus.isi.edu/workflow_gallery/).
- CYBERSHAKE: The Cybershake workflow is used in South California Earthquake Center (SCEC) to characterize earthquake hazards using the Probabilistic Seismic hazards analysis (PSHA) technique (https://pegasus.isi.edu/workflow_gallery/).
- MONTAGE: This generates the custom mosaics of the sky using input images in the Flexible Image Transport System (FITS) format (https://pegasus.isi.edu/workflow_gallery/).
- LIGO: This workflow is used in Einstein's theory of general relativity. The Laser Interferometer Gravitational-wave Observatory (LIGO) attempts to detect gravitational waves produced by various events in the universe (https://pegasus.isi.edu/workflow_gallery/).
- SIPHIT: This workflow is used in the automated untranslated search for RNAs from bioinformatics bacterial database projects at Harvard University (https://pegasus.isi.edu/workflow_gallery/).

3.3. Workflow Scheduler

To optimize defined objectives, a scheduler analyzes and distributes its tasks to the available resources. It provides a summary of the workflow, identifies multiple queues and distributes tasks to effectively run the device based on the user's requirements. Workflows can be forwarded to resources using simple approaches such as queues or more complicated techniques. When a user needs to use cloud resources efficiently, an efficient scheduler is required to facilitate his/her goals. The main function of workflow schedulers, such as GA, ACO and PSO described below, is to provide satisfactory QoS to end-users.

3.3.1. Genetic Algorithm (GA)

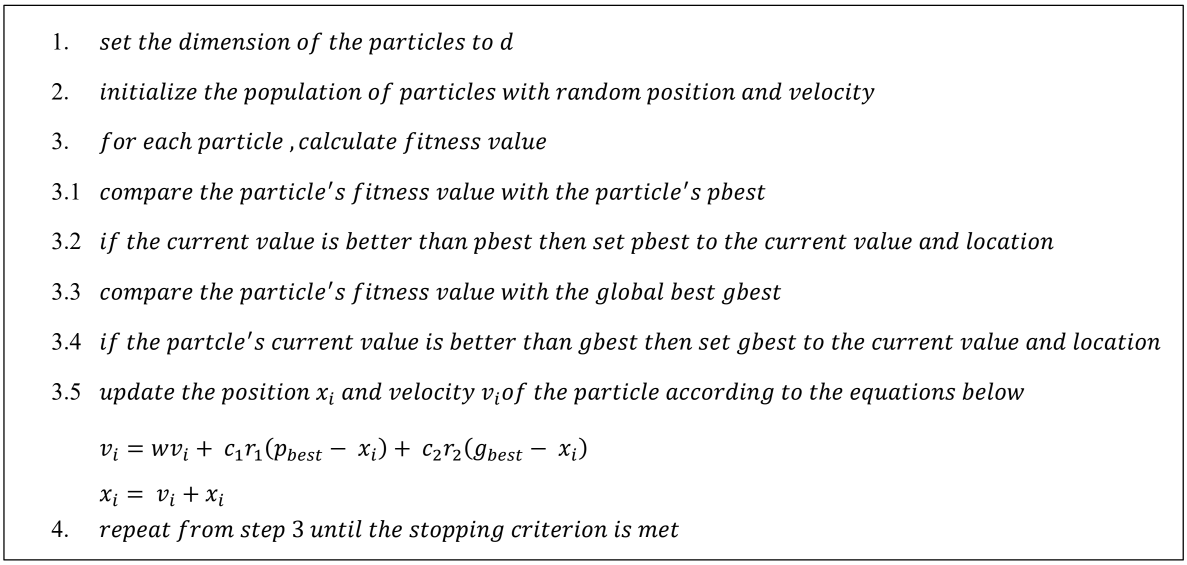
GA is a bioinspired algorithm developed in 1960 by John Holland from the University of Pennsylvania (USA). It is an optimization strategy that is well-known for finding an approximate solution to a search problem. It is used in many scientific applications such as cancer scanning, gene expression profiling analysis, robotics, telecommunications, engineering design, automotive and marketing [20].

3.3.2. Ant Colony Optimization (ACO)

ACO was proposed by Marco Dorigo in 1992. It is a probability-based algorithm for finding the best path. It simulates the procedure by which ants find the shortest path during their search for food. Practical applications of ACO include the train scheduling system, timetabling, shape optimization, telecommunication network design and problems in computational biology [21].

3.3.3. Particle Swarm Optimization (PSO)

PSO was developed by Eberhart and Kennedy in 1995. It is a population-based optimization technique that simulates the social behavior of birds flocking or fish schooling [17]. A typical PSO algorithm is presented in Figure 3. PSO can be applied in many different areas such as artificial neural network, training, function optimization, fuzzy system control and other areas where GA can be used [22].



```
1.    set the dimension of the particles to d
2.    initialize the population of particles with random position and velocity
3.    for each particle , calculate fitness value
3.1   compare the particle's fitness value with the particle's pbest
3.2   if the current value is better than pbest then set pbest to the current value and location
3.3   compare the particle's fitness value with the global best gbest
3.4   if the partcle's current value is better than gbest then set gbest to the current value and location
3.5   update the position $x_i$ and velocity $v_i$of the particle according to the equations below
      $v_i = wv_i + c_1r_1(p_{best} - x_i) + c_2r_2(g_{best} - x_i)$
      $x_i = v_i + x_i$
4.    repeat from step 3 until the stopping criterion is met
```

Figure 3. Pseudocode of the PSO algorithm.

Where

w = inertia;
c_i = acceleration coefficient, i = 1,2;
r_i = random number, i = 1,2 and $r_i \in [1,2]$;
x_i = current position of particle i;
p_{best} = best position of particle i and
g_{best} = position of the best particle in the population.

The three algorithms mentioned above have their ways of solving complex optimization problems. Each algorithm has its characteristic performance in finding the best solution, depending on the problems. They can be compared based on the differences in their operation.

3.4. QoS Constraints

In cloud computing, customers define QoS constraints according to their requirements. Some of these QoS constraints in the literature are described in Table 2 below.

Table 2. QoS constraints for workflow scheduling.

S. No.	Constraint	Description
1	Makespan	The period between the starting time of the execution and the completion time of the actual workflow [23].
2	Cost	The amount paid by users for executing workload on cloud providers' services [24].
3	Throughput	The total number of users' requests finished within a time limit [25].
4	Reliability	This is the ratio of the total number of performed tasks to the total number of tasks. The aim is to provide services to clients [26].
5	Resource utilization	The appropriate use of resources in the course of workflow scheduling using the idle time gaps [27].
6	Turnaround time	The difference between the completion time and the task submission time [28].
7	Success rate	The total number of workflows carried out within user-defined constraints [29].
8	Tardiness	This defines how long workflow schedule has been postponed to the extent that the time of completion exceeds the estimated time limit [30].
9	Resource availability	This estimates the number of resources available to map tasks in order to reduce the rate of failure [31].
10	Load balancing	This defines how the scheduler optimizes resources used to reduce the pressure of cloud resources [11].
11	Response time	The time between task arrival and task completion [32].
12	Budget	The expense of using cloud services for a certain period of time [33].
13	Deadline	The user's time limit to perform the workflow [34].
14	Waiting time	This determines the interval between the time the task is ready and when the task begins [35].
15	Execution time	The time it takes for the resource to perform the task [23].
16	Security	This describes a stable scheduling to reduce the effect of security attacks by attackers via abusing the cloud services [11].
17	Energy consumption	This determines the utilization of the energy during the scheduling process [36].
18	Fault tolerance	This identifies the hardware and software problems that can be occurred at the start of execution until the last job in the workflow is completed [37].

Meeting these requirements is a primary challenge in workflow scheduling. To address this issue, different workflow scheduling algorithms have been proposed. In the next section, we review research works on PSO-based workflow scheduling schemes that consider QoS constraints.

4. PSO-Based Scheduling

PSO has an advantage over evolutionary algorithms in which it has a rapid convergence capacity. Additionally, some shortcomings of PSO like local optima can be solved by changing some of its features and formulas or by applying PSO with other metaheuristic algorithms. Different categories of PSO-based scheduling algorithms [10] can be considered in diverse applications. They can also be used for scheduling a realistic workflow on virtual machines (VMs) in the cloud. Examples of these categories are standard PSO [38–41], jumping PSO [42], learning PSO [41], bi-objective PSO [43,44], modified PSO [45–52], binary PSO and its variants [16,53–57], and hybrid PSO [58]. Figure 4 below shows the classification of PSO-based algorithms in cloud computing. In what follows, we review and classify the 79 selected research articles in line with the earlier identified RQs.

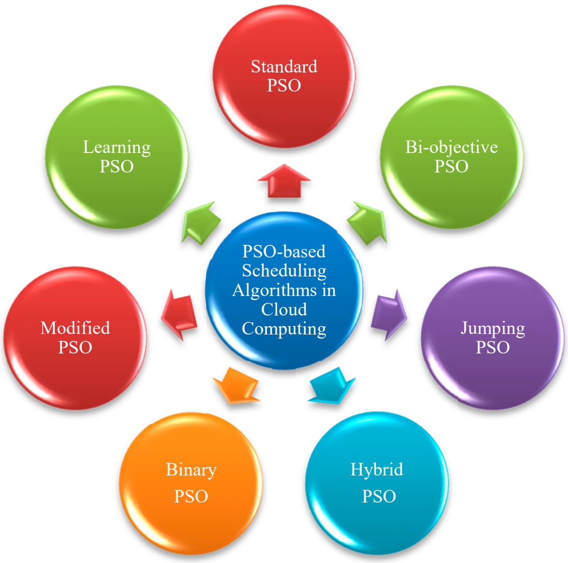


Figure 4. Taxonomy of PSO-based scheduling algorithms in cloud computing.

4.1. Standard PSO

Overcoming the drawbacks of cloud VM migration is costly and time-consuming. Instead of moving the entire overloaded VM, the PSO-based task scheduling algorithm [38] was suggested to shift the task from overloaded VMs to underloaded VMs. A new optimization model was developed to convert these new tasks into VM, optimize makespan as well as transfer time. The architecture of cloud resource brokers was designed, developed and built by [39]. Reference [59] states that controlling the resources and developing the different kinds of QoS parameters based on specified fitness function using the PSO algorithm is a constraint. PSO-based asset planning technique, called BULLET was suggested by [60] to reduce running costs, time, availability and power along with other QoS parameters. This suggested PSO-based algorithm has been used to efficiently plan resources to optimally solve the problem using the fitness function. The fitness function is more effective [61,62] in allocating the best resources for applications (tasks). It enables all applications to be processed in the shortest possible time at a minimum cost.

4.2. Jumping and Learning PSO

In the standard PSO technique, the global best particle, g_{best}, gets stuck in the local minima because it is not dynamically adjusted in all iterations which yield a poor convergence rate. To address this drawback, the Jumping PSO technique was proposed. This technique involves moving particulate matter from one coordinate to another and reducing workflow scheduling compilation time [42]. Self-adaptive learning PSO [41] incorporates four-velocity updating mechanisms for the IaaS cloud to delegate user tasks effectively and increase the revenue of cloud service providers.

4.3. Bi-Objective PSO

Bi-objective PSO is a variation of PSO that simultaneously optimizes two objectives in the cloud environment. This workflow takes deadline and budget constraints into account to optimize the costs and time of implementation. A priority bi-objective PSO algorithm was proposed by [43] to simultaneously optimize both cost and makespan. The proposed algorithm assigns the result of HEFT to initialize PSO. The simulations of four different workflow systems in the real-world and correlations with other algorithms determine the efficiency of the algorithm. The results of the computations indicate that the proposed algorithm performs better than others. To simultaneously optimize both parameters, i.e., time and costs, authors in [44] suggested the use of the smallest position value (using

PSO technique) to meet the end-user requirements and reduce infrastructure cost (thereby maximizing profit for cloud service providers).

4.4. Modified PSO (MPSO)

In the last decade, several modifications to PSO algorithms (known as MPSOs) have been proposed to optimize the performance of cloud computing for different QoS parameters. A lot of MPSOs now exist to overcome the weaknesses in the existing PSO algorithm. For example, [45] introduced an improved PSO planning algorithm to solve the cloud resource planning problem. The algorithm looks for the best resource for the next task and assigns the task to that resource to minimize completion time and cost. This is executed based on the current workload at VMs. Results showed that MPSO algorithms are better than the existing PSO algorithm in terms of time, cost, speed, and effectiveness. Similarly, [47] introduced an MPSO algorithm that optimized the fitness function to reduce the processing time and utilization of cloud resources. A new PSO methodology for the IaaS cloud, called the NPSO, was introduced in [48] to minimize the financial costs and time taken to finalize applications. An updated PSO algorithm was also proposed by [61] to address particle decline randomness and find an optimal global solution. The proposed technique provides one-to-one mapping and the fastest processor assignment tasking.

The MPSO techniques mentioned above are characterized by premature convergence and stagnation. Thus, efforts were made to resolve these problems. In this context, [50] proposed PSO algorithm distribution-dependent update rules that evaluate output at 13 non-linear global benchmark optimization functions. Experimental evidence shows that the proposed PSO-based algorithm optimizes the fitness function better than the existing algorithms. To increase the global search efficiency [62], an alternative modified APSO-VI algorithm was proposed based on the average absolute velocity of the evading particles. The experimental findings showed that the proposed algorithm escaped from a local minimum without premature convergence. [52] used the APSO-VI algorithms to schedule applications in a cloud environment. The proposed algorithm optimized different QoS parameters (like cost, time, throughput and energy consumption as well as task rejections) when compared to other state-of-the-art algorithms (such as PSO) considering the constraint of the time limit.

4.5. Binary PSO (BPSO)

Most real-world optimization problems are distinct. Examples of such include task scheduling, 0–1 knapsack problem and travel salesman problem. These problems can be solved using the BPSO algorithm. This binary version of PSO [16] was proposed for discrete optimization problems in 1997. The sigmoid transfer function was used to convert the velocity value from continuous to binary. BPSO has been used to solve diverse discrete optimization problems [53–55]. It has good convergence ability but it is affected by a lack of diversity in its premature convergence. An active research focus is to enhance BPSO's exploration and development capability. In this context, a sigmoid transmission, a linear transfer function, and two separate V-form transfer functions have been proposed to solve the problems of exploration-exploitation in BPSO [16,56,57].

4.6. Hybrid PSO

In conjunction with one or more scheduling algorithms, PSO can be used to solve several practical problems. Such combinations are named PSO hybrid algorithms. They can solve problems of local minimum, premature convergence, low convergence accuracy, etc. For example, to improve the completion time and resource usage ratio in cloud computing, resource planning algorithms [58] combine SA with PSO to form an improved PSO (IPSO) algorithm. SA increases the convergence speed and accuracy. This is achieved by adding PSO throughout the simulation process in every iteration. In IPSO algorithms, SA also increases PSO search speed. Krishnasamy and Gomathi et al. [63]

discussed an additional hybrid PSO (PSO and DE) algorithm to balance the workload and minimize cloud computing time for applications.

5. PSO-Based Workflow Scheduling Schemes

The particle swarm optimization algorithms for workflow scheduling can be generally categorized as heuristic, meta-heuristic and hybrid schemes (Figure 5). With regards to RQ1, it is observed that most researchers concentrate on the following techniques for planning workflows in distributed environments.

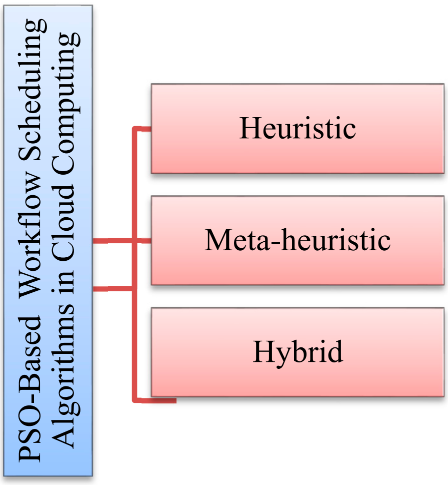


Figure 5. Classification of PSO-based workflow scheduling schemes.

5.1. Heuristic Algorithms

Heuristic means "to be found by trial and error." This group of algorithms solves optimization problems in a reasonable time. However, optimal solutions cannot be guaranteed. This is fine if we do not want the best solutions that can easily be found [64]. These algorithms were used by previous researchers to solve scheduling problems in cloud computing. For example, the approach in [65] implements service cost on PSO (PSO-SC) to optimize workflow in a dynamic cloud scenario. PSO-SC approach did not only reduce the computing time but also decrease the cost of running users' tasks during the scheduling process. Results show that the approach effectively schedules tasks and reduces the complexities associated with such scheduling processes.

Table 3. Summary of the reviewed PSO heuristic algorithm for workflow scheduling.

Year	Algorithm	Type of Algorithm	Type of Workflow	Testing Tool	Advantage	Disadvantage
2016 [65]	PSO-SC	Dynamic	DAG	Mathematical formula and simulation software (Cloudsim).	It improves the performance and provides a best optimal solution.	It is trapped in local optima.

Table 4. QoS parameters in the reviewed PSO heuristic algorithm.

Year	Reduce Execution Time	Cost	Energy Consumption	Fault Tolerance	Single/Multi Objective	Scheduling Strategy
2016 [65]	✓	✓	✗	✗	MO	DYNAMIC

Tables 3 and 4 present the current PSO-based heuristic algorithm used by researchers to address workflow scheduling problems. The tables provide comprehensive answers to RQ1, RQ2, RQ3 and

RQ4. They indicate the source of the algorithm, its advantages and disadvantages, the testing tool used in the experiment, target QoS constraints, etc.

5.2. Meta-Heuristic Algorithms

The word "meta" means "above" and usually, the meta-algorithms do much better than simple heuristics. This is because they involve randomization and local searches. Randomization provides a good way to escape local searches and thus all meta-heuristic algorithms are built for global optimization [64]. Next, we review some PSO-based meta-heuristic algorithms.

Pandey et al. [66] found PSO to be the most effective for run-time workflow scheduling. This is due to its low computation and communication cost. Also, there are two considerations for obtaining optimized solutions: one is the heuristic scheduling process while the other is the PSO for optimized performance "task-resource mapping". In [67], Wu et al. proposed an *RDPSO*-based PSO algorithm where each solution is described in task-set pairs. Greedy's Randomized Adaptive Research Process (GRASP) is used to maximize the initial swarm population. A three-stage process is then followed to establish new swarms. The "gbest" and "pbest" particles are picked at the first level. However, due to the discrete properties of scheduling, gbest pairs are not well-optimized in the next step for producing new locations as they 'learn' from their previous location. The unmapped tasks pick resources from other optimized pairs in the third step. The authors concluded that RDPSO surpasses PSO with respect to minimizing costs.

Thanh et al. [68] proposed a new version of the PSO algorithm that was proven to solve the problem of workflow preparation. The *PSOi* deploys other approaches to reach optimal solutions without being trapped in local optimum solutions. This version uses a new strategy for transferring particles into a new space called "inverse". After every iteration, PSOi helps to change the particulate position. A Catfish PSO (*C-PSO*) algorithm [19] was proposed to select the best task schedule with the least execution time and makespan. It was developed to schedule a large scientific workflow in an IaaS. As hypothesized, the algorithm was able to efficiently schedule tasks and map them to their appropriate resources. Another proposed workflow scheduling algorithm was named *PSO-DS* with CUPA features [18]. This algorithm uses a workflow manager system (WMS) to create a direct link between the workflow owners and resources. Thus, WMS was used with the required protocol to start communication between resources in the experiment.

Many of the proposed scheduling algorithms for cloud computing fall short of meeting the required QoS of users or do not take other basic principles of cloud computing such as heterogeneity of resources into consideration. Reference [69] proposed a resource scheduling strategy for scientific workflow in the IaaS cloud. The approach uses a PSO meta-heuristic algorithm to minimize the total execution cost of a workflow considering the deadline constraints. The main objective was to optimize the workflow scheduling in the cloud, considering the dynamism in IaaS resource provisioning and scheduling. This approach uses PSO not just for mapping tasks to resources but for determining the number and type of virtual machine to be leased and when to be leased and released. Moreover, it considers diverse IaaS cloud characteristics such as variation performance and resource boot time. The proposed solution merged different underlying aspects of IaaS cloud such as elasticity, heterogeneity, pay-by-use, and resource dynamism. The drawback of this approach is that it defines deadline constraints for resource provisioning and execution cost minimization.

Similarly, a multi-swarm multi-objective advanced operation algorithm (*MSMOOA*) [70] was proposed to improve multi-objective workflow process in cloud computing. The approach uses different kinds of swarms to cater to diverse issues, thereby enabling efficient data sharing among the swarms. Each physical machine works with these swarms. The swarms are later "upgraded" to a multi-objective molecule used for discovering a "non-commanded" arrangement as a single objective. MSMOOA groups particles of the swarm into two classes. Particles in the first category communicate with particles of different swarms to energize data sharing among the swarms. The second class of the particles trades the data between particles of similar swarms. In the approach, the server manager (SM) is used to maintain the effect of accessing resource visibility for the mapping. Experimental results comparing MSMOOA and two other approaches (HEFT and MOHEFT) indicated that MSMOOA performed significantly better than those approaches.

The authors in [71] developed a *Fuzzy PSO with LJFN and SJFN* algorithm using LJFN and SJFN heuristics. They used a method that swaps LJFN and SJFN heuristics when a new job is assigned to grid nodes. Nevertheless, the number of grid nodes allocated based on FCFS and LJFN is greater than the number of tasks. The proposed approach creates optimal solutions in time to reduce preparation time and increase resource use efficiency. [41], [72] suggested a PSO-based strategy that takes advantage of PSO's quick convergence. The proposed method (*SLPSO*) deploys various speed upgrade methods that prevent the solution from being stuck in the local optima and boost the performance. This involves other policies such as adaptive parameters changing, designing different population topologies, using multi-population in standard PSO and bio-inspired PSO methods that combine PSO with other adaptive systems. Within the user-specified QoS constraints, SLPSO performs efficiently well.

PSO with VNS was proposed by [73]. It combines four procedures: "initialization", "particle string update", "fitness calculation" and "solution selection". Before these procedures, the "particle string" must be generated to encrypt promising solutions. VNS was implemented to increase the reliability of the solutions in the solution selection process. On the same subject, Chen et al. [74] studied *S-CLPSO* to control the user-specified constraints using PSO and clarified the set-based PSO approach and its suitability for workflow scheduling. Speed and place are modified in every iteration in the S-CLPSO algorithm. The SPSO strategy tends to be a better option for workflow scheduling problems since service instances in the cloud are treated as a collection. Also, it is simple to 'expedite search' with S-PSO. It was therefore concluded that under very tight constraints, the findings obtained by S-CLPSO are very promising.

Tables 5 and 6 provide a comparative analysis of the various meta-heuristic algorithms used in the literature to manage workflow scheduling problems. The tables present a systematic annual review of several meta-heuristic algorithms indicating the source of each algorithm, its advantages and disadvantages, the testing tool used in the experiment, target QoS constraints, scheduling strategy, etc. Both tables answer RQs 1–4.

5.3. Hybrid Algorithms

This section reviews research works that combine heuristic and meta-heuristic algorithms. A systematic analysis of the various hybrid algorithms proposed for workflow scheduling is shown in Tables 7 and 8 based on the year of publication. These tables also provide answers to RQs 1–4 indicating the strengths and weaknesses of the algorithms. The testing tools used in the experiments, target QoS constraints, scheduling strategy, etc. are also highlighted.

Table 5. Summary of the reviewed PSO meta-heuristic algorithms for workflow scheduling.

Year [Ref]	Algorithm	Type of Algorithm	Type of Workflow	Testing Tool	Advantage	Disadvantage
2010 [66]	PSO	Dynamic	Simple	JSwarm6 package for simulation	PSO attains less cost than Best Resource Selection (BRS).	It does not consider real applications.
2010 [67]	RDPSO	Dynamic	Simple	simulation	RDPSO yields feasible solutions.	Not Specified.
2012 [74]	S-CLPSO	Static	Real-life application workflows and PSPLIB workflows	Simulation	S-CLPSO produce effective solutions while considering QoS constraints.	Not Specified.
2014 [43]	BPSO	Static	Scientific Workflows	Cloudsim	BPSO is better than PSO in terms of performance and cost.	They have not considered the load of resources.
2015 [68]	PSOi	Static	Simple	CloudSim, packet library Jswarm	PSOi is effective in terms of makespan in the small-scale cloud.	It has low performance when solving large instances in less execution time.
2017 [4]	HPSO	Dynamic	Scientific Workflows	Cloudsim	Minimizes the cost of execution and time simultaneously.	The consumption of energy and other effective QoS is not considered.
2014 [69]	PSO	Dynamic	Scientific Workflows	Cloudsim	It involves basic IaaS cloud concepts such as pay-as-you-go model, flexibility, elasticity, and resource dynamics.	It does not consider budget and reliability constraints.
2017 [70]	MSMOOA	Dynamic	Scientific Workflows	WorkflowSim	Multi-modulated particle optimization algorithm (MOPSOA) is used to find the non-dominated solutions with a single objective called a swarm in every PM.	Not Specified.
2016 [19]	C-PSO	Dynamic	Scientific Workflows	WorkflowSim	It yields significant changes in makespan and execution cost in comparison to PSO for 400 tasks.	C-PSO performance is slightly better than PSO for workflows of 100, 200 and 300 tasks.
2017 [18]	PSO-DS	Dynamic	Scientific Workflows	CUPA	It can produce better results in terms of cost and makespan. Specifically, for those with fewer resources that provide functional values over 80%.	PSO-DS needs a high budget to be implemented by users.
2017 [75]	DPSO	Dynamic	Scientific Workflows	WorkflowSim.	It schedules the medical workflow application with a discrete PSO.	Not Specified.
2019 [76]	DNCPSO	Dynamic	Scientific Workflows	WorkflowSim.	It effectively and efficiently deals with the workflow scheduling issue in cloud–edge environment to reduce the makespan and cost.	Not Specified.

Table 5. *Cont.*

Year [Ref]	Algorithm	Type of Algorithm	Type of Workflow	Testing Tool	Advantage	Disadvantage
2018 [77]	MAPSO	Dynamic	Scientific Workflows	Simple benchmark program	It minimizes the total execution time and cost of the workflow while meeting multiple QoS constraints.	The consumption of energy and fault tolerance are not considered.
2018 [78]	APMWSA	Dynamic	Real-life applications Workflows	Cloudsim	It runs the workflow execution process to minimize total cost and makespan. This algorithm uses the concept of the novel adaptive elite-based PSO (NAEB-PSO) for task resource mapping.	Not Specified.

Table 6. QoS parameters of the reviewed PSO meta-heuristic algorithms for workflow scheduling.

Year	Makespan	Cost	Execution Time	Reliability	Utilization	Response Time	Budget	Deadline	Throughput	Energy Consumption	Fault Tolerance	Single/Multi Objective	Scheduling Strategy
2010 [66]	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	SO	DYNAMIC
2010 [67]	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	MO	DYNAMIC
2012 [74]	✓	✓	✗	✓	✗	✗	✓	✓	✗	✗	✗	MO	STATIC
2014 [43]	✗	✓	✓	✗	✗	✗	✓	✓	✗	✗	✗	MO	STATIC
2015 [68]	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	SO	STATIC
2017 [4]	✓	✓	✗	✗	✗	✗	✓	✓	✗	✓	✗	MO	DYNAMIC
2014 [69]	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗	✗	MO	DYNAMIC
2017 [70]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✓	✗	MO	DYNAMIC
2016 [19]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2017 [18]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2017 [75]	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2019 [76]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2018 [77]	✗	✓	✓	✓	✗	✗	✓	✓	✗	✗	✗	MO	DYNAMIC
2018 [78]	✓	✓	✗	✗	✗	✗	✓	✓	✗	✗	✗	MO	DYNAMIC

Table 7. Summary of the reviewed PSO hybrid algorithms for workflow scheduling.

Year	Algorithm	Type of Algorithm	Type of Workflow	Testing Tool	Advantage	Disadvantage
2017 [22]	PSO with MIN-MAX	Dynamic	DAG	Java Programming	LAPSO algorithm efficiently handles various QoS constraints in terms of trustworthy networking environments and successful optimization of the users' QoS objectives.	Not Specified.
2009 [26]	RHDPSO	Static	DAG of the workflow in e-protein project	Java programming language	Shows convergence speed and ability to obtain faster and feasible schedules.	Performs well only in grid background.
2015 [79]	Hybrid PSO-ACO	Dynamic	Task workflow	Workflow method	It proposes a new framework for scheduling single tasks on the resource sets dynamically.	Not Specified.
2015 [80]	Hybrid PSO	Dynamic	Simple	Cloudsim	Performs better in terms of schedule length.	Not Specified.
2017 [81]	ACO with PSO	Dynamic	Scientific workflow	CloudSim	In addition to reduced time delay, ACO-PSO produces an efficient schedule with lower cost.	Data transfer cost between centres is not studied.
2015 [82]	SA with PSO	Dynamic	Simple	CloudSim	This method maximizes the resource utilization and minimizes the makespan.	Not Specified.
2017 [83]	SA with PSO	Dynamic	Workflow of 100 jobs	MATLAB	Experimental results showed that the proposed method outperforms the techniques available for various quality indicators.	Not Specified.
2017 [84]	PSO + GA-PSO	Dynamic	Simple DAG	CloudSim	It allows consumers to choose resources equally from different geographical sites, with a reduced time of execution. This decreases production costs in return.	Not Specified.
2013 [85]	PSO with HEFT	Static	Synthetic workflow application, neuro-science workflow and a protein annotation workflow	Grid toolkit	Improves makespan time, cost and energy consumption.	It does not consider any constraint like the deadline, priority of applications, etc.
2015 [86]	PBM with PSO.	Dynamic	Normal workflow DAG	Cloudsim	The ability of the PSO approach to explore the problem space has been improved by using random inertia weight to provide particles with the ability to find better solutions during the late stages of the search.	Not Specified.

Table 7. *Cont.*

Year	Algorithm	Type of Algorithm	Type of Workflow	Testing Tool	Advantage	Disadvantage
2017 [87]	Hybrid PSO+ACO	Dynamic	Scientific workflow	Cloudsim	It uses PSO and ACO hybrids for several purposes and minimizes the overall run-time and cost.	Not Specified.
2015 [88]	PSO with gravitation search algorithm	Dynamic	DAG	Cloudsim	In comparison to non-heuristic implementations, the results of the experiments indicated a a 30% decrease in cost than PSO. Also, a 30% cost reduction in comparison to the gravitational search algorithm was recorded.	Not Specified.

Table 8. QoS parameters of the reviewed PSO hybrid algorithms for workflow scheduling.

Year	Makespan	Cost	Execution Time	Reliability	Utilization	Response Time	Budget	Deadline	Throughput	Efficiency	Availability	Security	Reputation	Energy Consumption	Fault Tolerance	Single/Multi Objective	Scheduling Strategy
2017 [22]	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✓	✓	✓	✗	✗	MO	DYNAMIC
2009 [26]	✓	✓	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	STATIC
2015 [79]	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗	✗	✗	MO	DYNAMIC
2015 [80]	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	SO	DYNAMIC
2017 [81]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2015 [82]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2017 [83]	✓	✗	✗	✗	✓	✗	✗	✗	✗	✓	✗	✗	✗	✗	✗	MO	DYNAMIC
2017 [84]	✓	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2013 [85]	✓	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓	✗	MO	DYNAMIC
2015 [86]	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2017 [87]	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	MO	DYNAMIC
2015 [88]	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	SO	DYNAMIC

Authors of [79] introduced the *PSO-ACO* algorithm, which is a fusion of PSO and ACO algorithms. The algorithm focuses on reducing the cost and time expended in PSO for the "fitness test" and seeks the global optimal solutions in ACO. The initial approach is to initialize the population then compare them based on an iterative loop objective function. The steps are repeated by changing the speed and position of particles up to a full schedule. Moreover, ACO also uses the global updating process and job rescheduling. [80] proposed a new *PSO and TS* algorithm in which PSO executes global search and TS performs a local search. The idea behind this hybrid approach is to develop both local (in confined space) and global solutions. It finds viable solutions while preventing solutions from sticking to the local optima.

A non-dominance sort-based Hybrid Particle Swarm Optimization (*HPSO*) algorithm was introduced in [4] to solve scheduling problems with conflicting objectives in the IaaS cloud. HPSO is an extension of the authors' previously proposed Budget and Deadline Constrained heterogeneous Earliest Finish Time (BDHEFT) algorithm: a form of multi-objective PSO. One of the disadvantages of the proposed technology is that energy consumption was not considered during the scheduling phase. The HPSO aims to improve the deadline and budgetary constraints of two objectives: makespan and cost. In future works, the energy consumption of the created workflow scheduling could also be reduced while considering these two conflicting objectives. HPSO provides a collection of 'best solutions' a customer can choose from. Its operations consider non-predominant system arrangements to tackle the cloud work process for booking issues. It involves a mixture of a multi-objective PSO operation and rundown-based heuristics [4].

To improve the elasticity and heterogeneity of the existing works in the cloud for optimal scientific workflow scheduling in the IaaS cloud, [81] introduced a new meta-heuristic optimization strategy involving ACO and PSO algorithms. The hybrid finds local and global best solutions to minimize makespan and reduce cost. Another hybrid workflow scheduling algorithm was proposed in [82]. The approach combines the features of both PSO and SA. It was implemented on Cloudsim to improve on the brokers' services, reduce makespan as well as increase resource utilization.

Mapping tasks to the available resources in the cloud is quite challenging. Thus, a hybrid meta-heuristic algorithm for optimizing parallel scheduling processes in the cloud environment [83] was proposed. This improves on the hybridized SA and PSO algorithms using a crossover variation operator. The algorithm was able to effectively reduce the flow time and schedule length. Results from the experiment indicated that the approach outperforms some of the existing approaches. A PSO-based constraints-aware multi-QoS workflow scheduling strategy and a proposed look-ahead heuristics (*LAPSO*) [22] were used to provide QoS satisfaction for various end-users (EU) with diverse QoS objectives and optimization requirements. The strategy selects the best solution using the proposed constraints handling approach. It hybridizes PSO with a novel look-ahead mechanism based on min-max heuristics which improves the quality of choosing the best solution. Simulation results indicate that LAPSO guarantees the satisfaction of the EU constraints even in “tight” situations.

Another major issue in cloud industries is allocating and scheduling dynamic and virtual resources to the users for maximal profit. A multi-objective resource allocation (*GA-PSO*) algorithm was proposed in [84] to minimize cost, time, and energy consumption. The approach uses meta-heuristic algorithms to solve some scheduling issues encountered in cloud industries. PSO solves large optimization problems with superior search speed and GA solves both non-linear and highly complicated engineering problems. Experimental results indicated that GA-PSO was able to reduce cost and makespan and also increase resource utilization.

Conflicting constraints such as budget and deadline emerge in the course of cloud resources scheduling because cheaper resources are slower than expensive resources. Most of the previous studies concentrate on one objective, i.e., either time minimization or cost minimization, under the influence of user-specified QoS constraints. Because of the complexity of workflow and the dynamic nature of the cloud, a trade-off is required to make a balance between total time of execution and processing cost. Another effort, *RHDPSO* [26], showed that premature convergence and position maximum should be prevented. To this end, two methods were presented: first, "the discretization process" which is used to overcome the multi-QoS workflow constraint scheduling problem and second, "the random time sequence method" which can interrupt double particle extremities and solve premature convergence and local optimum problems. However, the regular PSO algorithms are superior to this hybrid form.

6. Summary of the Literature Review

This section summarizes the QoS metrics used in the reviewed literature for the evaluation of PSO-based workflow scheduling strategies. The rate at which each QoS metric is utilized for evaluation purposes is presented in Figure 6. The limitations of this review are also highlighted.

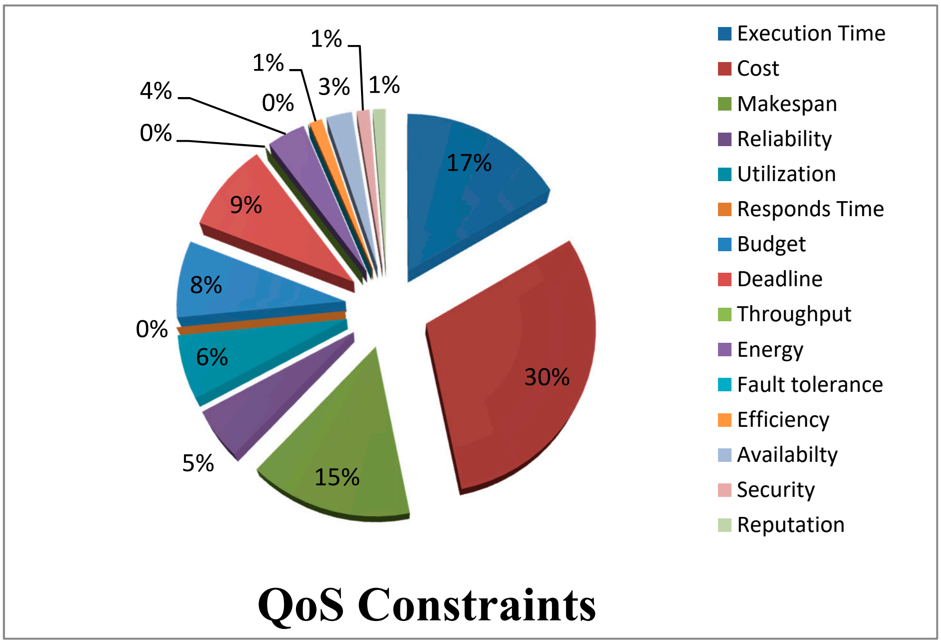


Figure 6. The percentages of QoS metrics considered in workflow scheduling strategies.

6.1. Percentages of QoS Metrics Used in Workflow Scheduling Strategies

From Figure 6, the most widely used evaluation measure by the researchers in the reviewed literature to evaluate PSO-based workflow scheduling strategies is cost. 30% of the proposed workflow scheduling strategies considered the execution cost. The metric with the second-highest percentage of usage (17%) is execution time followed by Makespan (15%). Resource utilization and reliability constraint metrics of PSO-based workflow scheduling rates represent 6% and 5%, respectively. Next to meeting users' defined deadline constraints (9%), energy consumption was used for evaluation by 4% of the total research on workflow scheduling encountered in this review. Efficiency, security, and reputation (1% each) were scarcely used.

6.2. Limitations of This Literature Review

Upon analyzing the data obtained from the literature review related to workflow scheduling, we realize the following limitations:

- The best criteria or methods for different databases were not defined.
- The accuracy of the algorithms has not been established.
- Not all the QoS constraints, e.g., load balancing, were addressed.

6.3. Historical Distribution

This paper presents the distribution of published research on workflow scheduling in the last few years. Specific RQs are considered for resolving the gaps in current strategies (RQ3). From Sections 1–6, we evaluated potential expectations (RQ5) after conducting QA, SDS, and DCP in the corresponding publications between the years 2000 to 2019.

6.4. Distribution of Publications per Year

The papers published in the years between 2000 and 2019 are shown in Figure 7. This analysis includes all articles that we got from all databases before the exclusion process. Figure 7 shows that one article was published in 2000 and 2001, 19 articles were published in 2010, 23 articles in 2012, 53 articles in 2015, 79 articles in 2017 and 47 articles in 2019.

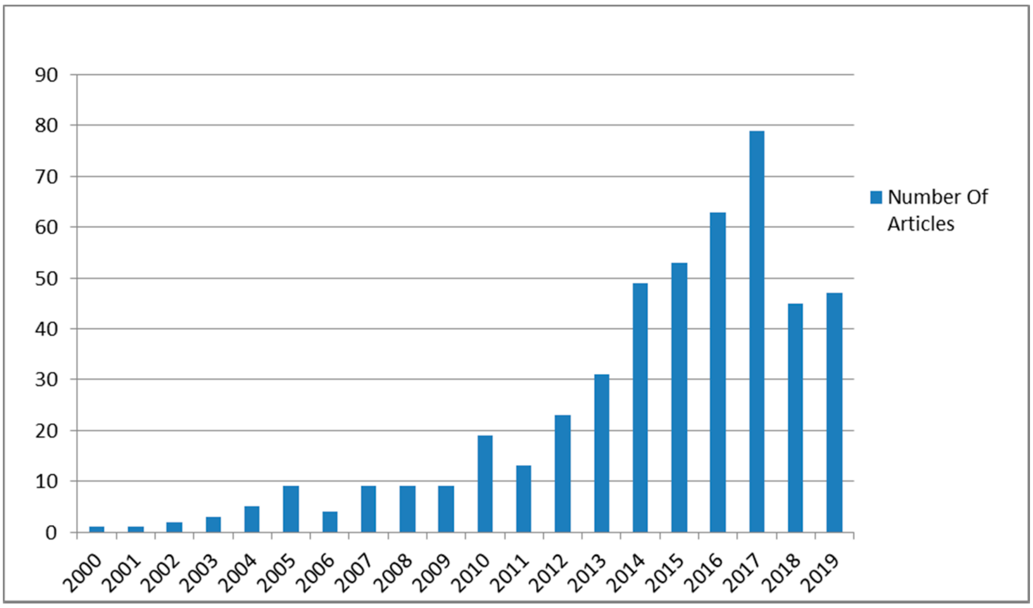


Figure 7. Distribution by publication year.

The highlighted measurements in most of the research articles reviewed within the scope of this paper (i.e., scheduling scientific workflow using PSO-based techniques) are illustrated in Figure 8. Researchers in this area considered 15 common metrics. The frequency count and the number of papers that used each metric are also provided in Figure 8. The important and least considered measurements are shown as well. The highlighted measurements i.e., Execution time, Makespan and Cost, have the highest frequency count. Fault tolerance, throughput, response time, reputation, efficiency and security are much less utilized in PSO-based scientific workflow scheduling. A classification of the studied QoS metric for each paper is also provided in Figure 8. Therefore, we note that Execution time, Makespan and Cost are the main concerns for most authors.

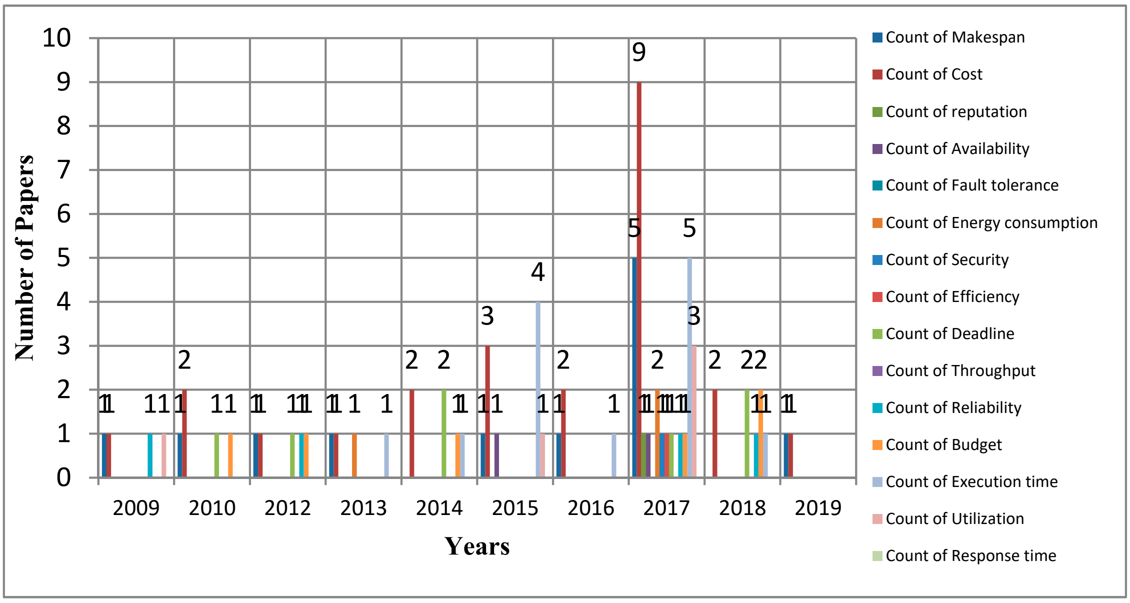


Figure 8. Number of studied metrics in the reviewed articles.

6.5. Research Validity

This research has carefully examined the existing literature. Nonetheless, some primary studies might not have been reviewed considering that researchers use various synonyms in the course of presenting their work. Moreover, we have thoroughly reviewed the techniques during the DCP stage to prevent the biased study selection problem.

7. Technical Comparison of Cloud, Fog and Edge Computing

The fog computing architecture consists of fog clusters wherein multiple fog devices cooperate. In contrast, the most important physical components of clouds are data centers. As a result, cloud computing is expensive to operate and it consumes energy. On the other hand, energy consumption and operation costs are low in the fog computing paradigm. Fog is closer to the user, so there can be one or a few hops between users and fog devices [89]. However, there is a significantly higher distance between users and the cloud. For this reason, the latency of communication for the cloud is higher when compared to fog. Cloud is more centralized while fog is regionally coordinated and has dispersed solution [90].

In edge computing, different platforms (all having different runtimes) could be used for programming. Cloud computing typically uses one programming language for one target platform. Edge computing requires a comprehensive security plan to address its state-of-the-art authentication and proactive attacks while the cloud does not require a massive security plan. Edge computing is known to be suitable for operations with very high latency requirements. Hence, medium-sized businesses with budget constraints can save financial resources using edge computing. On the other hand, cloud computing is better suited for large data storage programs and organizations [91]. The technical comparison between these three types of computing is shown in Table 9.

Table 9. Technical comparison of cloud, fog and edge computing.

Parameters	Cloud Computing	Fog	Edge
Internal connectivity	Mostly wired	Mostly wireless	Mostly wireless
Power source	Direct power	Battery, direct power, green energy such as solar power.	Battery, direct power, green energy such as solar power.
Power consumption	High	Low	Low
Computation capacity	High	Low	Low
Storage capacity	High	Low	Low
Participating node	variable	Constantly dynamic	Constantly dynamic
Management	centralized	Distributed/ centralized	Distributed
Computation device	Powerful server system	Any device with computation power	Any device with computation power
Nature of failure	predictable	Highly diverse	Highly diverse
Connectivity from user	High speed (with combination of wire and wireless)	Mostly wireless	Mostly wireless
Network latency	High	Low	Low
Node mobility	Very low	High	High
Number of intermediate hops	Multi	One/few	Single hop
Application type	Non latency-aware	latency-aware	latency-aware
Real time application handling	Difficult	Achievable	Achievable
Computation cost	High	Low	Low
Cooling cost	High	Very low	Low

8. Open Challenges and Future Research Direction

This section answers RQ5 by providing insights into further research directions. We recommend that future works should explore different choices that affect the performance of the scheduling algorithm in terms of: (1) selecting the initial resource pool which has a significant effect on the process, (2) using different optimization algorithms such as genetic algorithm and (3) comparing their performances with PSO. There is also a need to work with these algorithms to ensure the process of mapping tasks to resources takes place with enough memory size. Moreover, these algorithms should be implemented in workflow engines so that they can be used in different real-life applications. Introducing hybrid meta-heuristic algorithms can also improve the performance in the cloud.

Similarly, it is necessary that multi-objective solutions are introduced to workflow scheduling processes. Due to the vulnerability of the cloud environment to failure, fault-tolerant approaches are needed to ensure effective communication between users. Also, error-tolerant methods with lower complexity must be taken into account for designing multi-objective workflows. To reduce the energy consumption of cloud data centers and achieve green-cloud computing, more attention must be paid to objectives like the VM load balancing in the data center network. Furthermore, some recent cloud schedulers report security threats that negatively impact the quality of cloud services. This is generally because cloud systems overlook security problems. To either prevent or minimize the impacts of these security threats, future research addressing scheduling problems should focus on various factors relating to protection and workflow scheduling solutions.

9. Conclusions

Cloud computing is a new technology that gives the industry the ability to take the benefits of virtual resources on a pay-per-use basis. Its scheduling process involves mapping the tasks to the VMs to reduce the makespan and execution cost. Scheduling also enhances resource availability and system scalability for cloud providers thereby reducing the operational cost of data centres. A popular "unorganized" optimization technique for low computational and cost-effective applications suitable for workflow scheduling in cloud computing is Particle Swarm Optimization. In this respect, this paper presented a clear analysis of different PSO-based algorithms in cloud computing. This was done in line with the objectives of solving workflow scheduling problems. We note that future work should focus on scheduling workflows in a heterogeneous cloud environment. Also, the dynamic request for hybrid resources should be evaluated while considering different levels of reliability. Furthermore, scheduling algorithms should also cater to the trust concerns of cloud users who submit tasks for execution in the cloud.

Author Contributions: Writing original draft, M.F.; Writing review and editing, M.F.; Supervision and review, R.L.; Review, M.H. and N.A.W.A.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by Putra Grant, University Putra Malaysia (Grant No: 95960000) and the Ministry of Education Malaysia.

Acknowledgments: The authors would like to acknowledge the financial support and facilities provided by Universiti Putra Malaysia for the execution, completion and publication of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hu, H.H.; Li, Z.; Hu, H.; Chen, J.; Ge, J.; Li, C.; Chang, V. Multi-objective scheduling for scientific workflow in multicloud environment. *J. Netw. Comput. Appl.* **2018**, *114*, 108–122. [CrossRef]
2. Buyya, R.; Yeo, C.S.; Venugopal, S.; Broberg, J.; Brandic, I. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.* **2009**, *25*, 599–616. [CrossRef]
3. Chen, W.; da Silva, R.F.; Deelman, E.; Sakellariou, R. Balanced Task Clustering in Scientific Workflows. In Proceedings of the 2013 IEEE 9th International Conference on e-Science, Beijing, China, 22–25 October 2013; pp. 188–195.
4. Verma, A.; Kaushal, S. A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling. *Parallel Comput.* **2017**, *62*, 1–19. [CrossRef]
5. Shimpy, E.; Sidhu, J. Different scheduling algorithms in different cloud environment. *Int. J. Adv. Res. Comput. Commun. Eng.* **2014**, *3*, 8003–8006.
6. Kumar, M.; Sharma, S.C.; Goel, A.; Singh, S.P. A comprehensive survey for scheduling techniques in cloud computing. *J. Netw. Comput. Appl.* **2019**, *143*, 1–33. [CrossRef]
7. Jafarpour, B.M.; Shiry, R.M. A Hybrid Method for Optimization (Discrete PSO + CLA). In Proceedings of the International Conference on Intelligent and Advanced Systems 2007, Kuala Lumpur, Malaysia, 25–28 November 2007.
8. Verma, A.; Kaushal, S. Cost Minimized PSO based Workflow Scheduling Plan for Cloud Computing. *Int. J. Inf. Technol. Comput. Sci.* **2015**, *7*, 37–43. [CrossRef]
9. Bhatt, K.; Bundele, M. Review Paper on PSO in workflow scheduling and Cloud Model enhancing Search mechanism in Cloud Computing. *IJIET Int. J. Innov. Eng. Technol.* **2013**, *2*, 68–74.
10. Masdari, M.; Salehi, F.; Jalali, M.; Bidaki, M. A Survey of PSO-Based Scheduling Algorithms in Cloud Computing. *J. Netw. Syst. Manag.* **2017**, *25*, 122–158. [CrossRef]
11. Masdari, M.; ValiKardan, S.; Shahi, Z.; Azar, S.I. Towards workflow scheduling in cloud computing: A comprehensive analysis. *J. Netw. Comput. Appl.* **2016**, *66*, 64–82. [CrossRef]
12. Adhikari, M.; Amgoth, T.; Srirama, S.N. A survey on scheduling strategies for workflows in cloud environment and emerging trends. *ACM Comput. Surv.* **2019**, *52*, 1–36. [CrossRef]
13. Ala'Anzy, M.; Othman, M. Load Balancing and Server Consolidation in Cloud Computing Environments: A Meta-Study. *IEEE Access* **2019**, *7*, 141868–141887. [CrossRef]
14. Dawood, K.A.; Sharif, K.Y.; Zaidan, A.A.; Ghani, A.A.A.; Zulzalil, H.B.; Zaidan, B.B. Mapping and Analysis of Open Source Software (OSS) Usability for Sustainable OSS Product. *IEEE Access* **2019**, *7*, 65913–65933. [CrossRef]

15. Ali, A.Q.; Sultan, A.B.M.; Ghani, A.A.A.; Zulzalil, H. A Systematic Mapping Study on the Customization Solutions of Software as a Service Applications. *IEEE Access* **2019**, *7*, 88196–88217. [CrossRef]
16. Kennedy, J.; Eberhart, R.C. Discrete binary version of the particle swarm algorithm. *Proc. IEEE Int. Conf. Syst. Man Cybern.* **1997**, *5*, 4104–4108.
17. Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the 6th International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; Volume 267, pp. 39–43.
18. Casas, I.; Taheri, J.; Ranjan, R.; Zomaya, A.Y. PSO-DS: A scheduling engine for scientific workflow managers. *J. Supercomput.* **2017**, *73*, 3924–3947. [CrossRef]
19. Nirmala, S.J.; Bhanu1, S.M.S. Catfish-PSO based scheduling of scientific workflows in IaaS cloud. *Computing* **2016**, *101*, 11404–11409. [CrossRef]
20. Kalra, M.; Singh, S. A review of metaheuristic scheduling techniques in cloud computing. *Egypt. Inform. J.* **2015**, *16*, 275–295. [CrossRef]
21. Pei, Y.; Wang, W.; Zhang, S. Basic ant colony optimization. In Proceedings of the International Conference on Computer Science and Electronics Engineering, Hangzhou, China, 23–25 March 2012; Volume1, pp. 665–667.
22. Ambursa, F.U.; Latip, R.; Abdullah, A.; Subramaniam, S. A particle swarm optimization and min–max-based workflow scheduling algorithm with QoS satisfaction for service-oriented grids. *J. Supercomput.* **2017**, *73*, 2018–2051. [CrossRef]
23. Blythe, J.; Jain, S.; Deelman, E.; Gil, Y.; Vahi, K.; Mandal, A.; Kennedy, K. Task scheduling strategies for workflow-based applications in grids. In Proceedings of the IEEE International Symposium on Cluster Computing and the Grid 2005, Cardiff, Wales, UK, 9–12 May 2005; Volume 2, pp. 759–767.
24. Yu, J.; Buyya, R.; Tham, C.K. Cost-based scheduling of scientific workflow applications on utility grids. In Proceedings of the 1st International Conference on e-Science and Grid Computing (e-Science'05), Melbourne, Australia , 5–8 July 2005; Volume 2005, pp. 140–147.
25. Zhu, M.; Wu, Q.; Zhao, Y. A cost-effective scheduling algorithm for scientific workflows in clouds. In Proceedings of the 2012 IEEE 31st International Performance Computing and Communications Conference (IPCCC), Austin, TX, USA, 1–3 December 2012; pp. 256–265.
26. Tao, Q.; Chang, H.; Yi, Y.; Gu, C.; Yu, Y. QoS constrained grid workflow scheduling optimization based on a novel PSO algorithm. In Proceedings of the 8th International Conference on Grid and Cooperative Computing, Lanzhou, China, 27–29 August 2009; pp. 153–159.
27. Tsai, Y.L.; Liu, H.C.; Huang, K.C. Adaptive dual-criteria task group allocation for clustering-based multi-workflow scheduling on parallel computing platform. *J. Supercomput.* **2015**, *71*, 3811–3831. [CrossRef]
28. Yu, Z.; Shi, W. A planner-guided scheduling strategy for multiple workflow applications. In Proceedings of the 2008 International Conference on Parallel Processing—Workshops, Portland, OR, USA, 8–12 September 2008; pp. 1–8.
29. Zheng, W.; Xu, C.; Bao, W. Online Scheduling of Multiple Deadline-Constrained Workflow Applications in Distributed Systems. In Proceedings of the 2015 Third International Conference on Advanced Cloud and Big Data, Yangzhou, China, 30 October–1 November 2015.
30. Rimal, B.P.; Maier, M. Workflow Scheduling in Multi-Tenant Cloud Computing Environments. *IEEE Trans. Parallel Distrib. Syst.* **2017**, *28*, 290–304. [CrossRef]
31. Tao, Y.; Jin, H.; Wu, S.; Shi, X.; Shi, L. Dependable Grid Workflow Scheduling Based on Resource Availability. *J. Grid Comput.* **2013**, *11*, 47–61. [CrossRef]
32. Richard, P.; Cottet, F.; Richard, M. On-line scheduling of real-time distributed computers with complex communication constraints. In Proceedings of the 7th IEEE International Conference on Engineering of Complex Computer Systems, Skovde, Sweden, 11–13 June 2001; pp. 26–34.
33. Shi, J.; Luo, J.; Dong, F.; Zhang, J. A budget and deadline aware scientific workflow resource provisioning and scheduling mechanism for cloud. In Proceedings of the IEEE 18th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Hsinchu, Taiwan, 21–23 May 2014; pp. 672–677.
34. Anwar, N.; Deng, H. Elastic Scheduling of Scientific Workflows under Deadline Constraints in Cloud Computing Environments. *Future Internet* **2018**, *10*, 5. [CrossRef]
35. Yan, Y.; Chapman, B. Scientific Workflow Scheduling in Computational Grids—Planning, Reservation, and Data/Network-Awareness. In Proceedings of the 8th IEEE/ACM International Conference on Grid Computing, Austin, TX, USA, 19–21 September 2007; pp. 18–25.

36. Singh, S.; Chana, I. QoS-Aware Autonomic Resource Management in Cloud Computing: A Systematic Review Sukhpal. *ACM Comput. Surv.* **2016**, *48*. [CrossRef]
37. Smara, M.; Aliouat, M.; Pathan, A.S.K.; Aliouat, Z. Acceptance Test for Fault Detection in Component-based Cloud Computing and Systems. *Future Gener. Comput. Syst.* **2017**, *70*, 74–93. [CrossRef]
38. Kuila, P.; Jana, P.K. Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Eng. Appl. Artif. Intell.* **2014**, *33*, 127–140. [CrossRef]
39. Somasundaram, T.S.; Govindarajan, K. Cloudrb: A framework for scheduling and managing High-Performance Computing (HPC) applications in science cloud. *Future Gener. Comput. Syst.* **2014**, *34*, 47–65. [CrossRef]
40. Gill, S.S.; Chana, I.; Singh, M.; Buyya, R. Chopper: An intelligent QoS-aware autonomic resource management approach for cloud computing. *Clust. Comput.* **2017**, *21*, 1–39. [CrossRef]
41. Zuo, X.; Zhang, G.; Tan, W. Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 564–573. [CrossRef]
42. Chitra, S.; Madhusudhanan, B.; Sakthidharan, G.R.; Saravanan, P. Local minima jump PSO for workflow scheduling in cloud computing environments. *Adv. Comput. Sci. Appl.* **2014**, *279*, 1225–1234.
43. Verma, A.; Kaushal, S. Bi-Criteria Priority based Particle Swarm Optimization workflow scheduling algorithm for cloud. In Proceedings of the Recent Advances in Engineering and Computational Sciences (RAECS), Chandigarh, India, 6–8 March 2014; pp. 6–8.
44. Beegom, A.S.A.; Rajasree, M.S. A particle swarm optimization based pareto optimal task scheduling in cloud computing. *Lect. Notes Comput. Sci.* **2014**, *8795*, 79–86.
45. Elmana, Z.T.A.; Zakria, M.; Omara, F.A. Pso Optimization algorithm for Task Scheduling on The Cloud Computing Environment. *Int. J. Comput. Technol.* **2014**, *13*, 4886–4897. [CrossRef]
46. Chenhong, Z.; Shanshan, Z.; Qingfeng, L.; Jian, X.; Jicheng, H. Independent tasks scheduling based on genetic algorithm in cloud computing. In Proceedings of the 5th International Conference on Wireless Communications, Networking and Mobile Computing, Beijing, China, 24–26 September 2009; pp. 1–4.
47. Zhao, G. Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment. *Int. J. Grid Distrib. Comput.* **2014**, *7*, 33–42. [CrossRef]
48. Pragaladan, R.; Maheswari, R. Improve Workflow Scheduling Technique for Novel Particle Swarm Optimization in Cloud Environment. *Int. J. Eng. Res. Gen. Sci.* **2014**, *2*, 675–680.
49. Navimipour, N.J.; Milani, F.S. Task Scheduling in the Cloud Computing Based on the Cuckoo Search Algorithm. *Int. J. Modeling Optim.* **2015**, *5*, 44–47. [CrossRef]
50. Kiran, M.S. Particle swarm optimization with a new update mechanism. *Appl. Soft Comput. J.* **2017**, *60*, 670–678. [CrossRef]
51. Rao, J.; Wei, Y.; Gong, J.; Xu, C.Z. QoS guarantees and service differentiation for dynamic cloud applications. *IEEE Trans. Netw. Serv. Manag.* **2013**, *10*, 43–55.
52. Kumar, M.; Sharma, S.C. PSO-COGENT: Cost and energy efficient scheduling in cloud environment with deadline constraint. *Sustain. Comput. Inform. Syst.* **2018**, *19*, 147–164. [CrossRef]
53. Naeem, M.; Pareek, U.; Lee, D.C. Swarm intelligence for sensor selection problems. *IEEE Sens. J.* **2012**, *12*, 2577–2585. [CrossRef]
54. Lin, Y. Based on particle swarm optimization algorithm of cloud computing resource scheduling in mobile internet. *Int. J. Grid Distrib. Comput.* **2016**, *9*, 25–34. [CrossRef]
55. Han, L.; Huang, C.; Zheng, S.; Zhang, Z.; Xu, L. Vanishing point detection and line classification with BPSO. *Signal Image Video Process.* **2017**, *11*, 17–24. [CrossRef]
56. Bansal, J.C.; Deep, K. A modified binary particle swarm optimization for Knapsack problems. *Appl. Math. Comput.* **2012**, *218*, 11042–11061. [CrossRef]
57. Mirjalili, S.; Lewis, A. S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization. *Swarm EVolume Comput.* **2013**, *9*, 1–14. [CrossRef]
58. Zhan, S.; Huo, H. Improved PSO-based Task Scheduling Algorithm in Cloud Computing. *J. Inf. Comput. Sci.* **2012**, *13*, 3821–3829.
59. Ramezani, F.; Lu, J.; Hussain, F.K. Task-based system load balancing in cloud computing using particle swarm optimization. *Int. J. Parallel Program.* **2014**, *42*, 739–754. [CrossRef]

60. Gill, S.S.; Buyya, R. Resource Provisioning Based Scheduling Framework for Execution of Heterogeneous and Clustered Workloads in Clouds: From Fundamental to Autonomic Offering. *J. Grid Comput.* **2019**, *17*, 385–417. [CrossRef]
61. Al-maamariand, A.; Omara, F.A. Task Scheduling Using PSO Algorithm in Cloud Computing Environments. *Int. J. Grid Distrib. Comput.* **2015**, *8*, 245–256. [CrossRef]
62. Xu, G. An adaptive parameter tuning of particle swarm optimization algorithm. *Appl. Math. Comput.* **2013**, *219*, 4560–4569. [CrossRef]
63. Gomathi, B.; Krishnasamy, K. Task scheduling algorithm based on Hybrid Particle Swarm Optimization in cloud computing environment. *J. Theor. Appl. Inf. Technol.* **2013**, *55*, 33–38.
64. Yang, X. *Nature-Inspired Metaheuristic Algorithms*, 2nd ed.; Luniver press: Bristol, UK, 2010.
65. Xue, S.; Shi, W.; Xu, X. A Heuristic Scheduling Algorithm based on PSO in the Cloud Computing Environment. *Int. J. u- e- Serv. Sci. Technol.* **2016**, *9*, 349–362. [CrossRef]
66. Pandey, S.; Wu, L.; Guru, S.M.; Buyya, R. 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, Perth, Australia, 20–23 April 2010; pp. 400–407.
67. Wu, Z.; Ni, Z.; Gu, L.; Liu, X. A revised discrete particle swarm optimization for cloud workflow scheduling. In Proceedings of the International Conference on Computational Intelligence and Security, Nanning, China, 11–14 December 2010; pp. 184–188.
68. Thanh, T.P.; The, L.N.; Doan, C.N. A novel workflow scheduling algorithm in cloud environment. In Proceedings of the 2nd National Foundation for Science and Technology Development Conference on Information and Computer Science (NICS), Ho Chi Minh City, Vietnam, 16–18 September 2015; pp. 125–129.
69. Rodriguez, M.A.; Buyya, R. Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds. *IEEE Trans. Cloud Comput.* **2014**, *2*, 222–235. [CrossRef]
70. Yao, G.S.; Ding, Y.S.; Hao, K.R. Multi-objective workflow scheduling in cloud system based on cooperative multi-swarm optimization algorithm. *J. Cent. South Univ.* **2017**, *24*, 1050–1062. [CrossRef]
71. Liu, H.; Abraham, A.; Hassanien, A.E. Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. *Future Gener. Comput. Syst.* **2010**, *26*, 1336–1343. [CrossRef]
72. Zhang, G.; Zuo, X. Deadline constrained task scheduling based on standard-PSO in a hybrid cloud. *Lect. Notes Comput. Sci.* **2013**, *7928*, 200–209.
73. Netjinda, N.; Sirinaovakul, B.; Achalakul, T. Cost optimal scheduling in IaaS for dependent workload with particle swarm optimization. *J. Supercomput.* **2014**, *68*, 1579–1603. [CrossRef]
74. Chen, W.N.; Zhang, J. A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints. In Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), Seoul, Korea, 14–17 October 2012; pp. 773–778.
75. Prathibha, S.; Latha, B.; Suamthi, G. Particle swarm optimization based workflow scheduling for medical applications in cloud. *Biomed. Res.* **2017**, *1*, 380–385.
76. Xie, Y.; Wang, Y.; Yang, Y. A novel directional and non-local-convergent particle swarm optimization based workflow scheduling in cloud–edge environment. *Future Gener. Comput. Syst.* **2019**, *97*, 361–378. [CrossRef]
77. Adhikari, M.; Amgoth, T. Multi-Objective Accelerated Particle Swarm Optimization Technique for Scientific workflows in IaaS cloud. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Bangalore, India, 19–22 September 2018; pp. 1448–1454.
78. Kothyari, Y.; Singh, A. A multi-objective workflow scheduling algorithm for cloud environment. In Proceedings of the 3rd International Conference on Internet of Things: Smart Innovation and Usages (IoT-SIU), Bhimtal, India, 23–24 February 2018; pp. 1–6.
79. George, S. Truthful Workflow Scheduling in Cloud Computing Using Hybrid PSO-ACO. In Proceedings of the International Conference on Developments of E-Systems Engineering (DeSE), Duai, UAE, 13–14 December 2015; pp. 60–64.
80. Sridhar, M.; Babu, G.R.M. Hybrid Particle Swarm Optimization scheduling for cloud computing. In Proceedings of the IEEE International Advance Computing Conference (IACC), Banglore, India, 12–13 June 2015; pp. 1196–1200.
81. Goyal, M.; Aggarwal, M. Optimize Workflow Scheduling Using Hybrid Ant Colony Optimization (ACO) & Particle Swarm Optimization (PSO) Algorithm in Cloud Environment. *Int. J. Adv. Res. Ideas Innov. Technol.*

2017, *3*. Available online: https://www.semanticscholar.org/paper/Optimize-Workflow-Scheduling-Using-Hybrid-Ant-(ACO)-Rai-Lala/d7d2fbb96583d6c47f26f6b83dc09d26b478dbc5 (accessed on 22 January 2020).

82. Hussein, A.-O.; Green, S.R.C.; Alam, M. Cloudlet Scheduling with Population Based Metaheuristics. *J. Comput. Sci. Manag. Res.* **2015**, *1*, 348–351.

83. Sandhu, S.K. Hybrid Meta-heuristics based scheduling technique for Cloud Computing Environment. *Int. J. Adv. Res. Comput. Sci.* **2017**, *8*, 1457–1465.

84. Brintha, N.C.; Benedict, S.; Jappes, J.T.W.W. A bio-inspired hybrid computation for managing and scheduling virtual resources using cloud concepts. *Appl. Math. Inf. Sci.* **2017**, *11*, 565–572. [CrossRef]

85. Yassa, S.; Chelouah, R.; Kadima, H.; Granado, B. Multi-objective approach for energy-aware workflow scheduling in cloud computing environments. *Sci. World J.* **2013**, *2013*. [CrossRef] [PubMed]

86. Al-Olimat, H.S.; Alam, M.; Green, R.; Lee, J.K. Cloudlet scheduling with particle swarm optimization. In Proceedings of the 5th International Conference on Communication Systems and Network Technologies, Gwalior, India, 4–6 April 2015; pp. 991–995.

87. June, M.; No, I. Bi-objective Hybrid Particle Swarm Optimization & Ant Colony Optimization Workflow Scheduling Algorithm for Cloud. *Int. J. Adv. Res. Comput. Sci. Res.* **2017**, *8*, 2321–2327.

88. Mirzayi, S.; Rafe, V. A hybrid heuristic workflow scheduling algorithm for cloud computing environments. *J. Exp. Theor. Artif. Intell.* **2015**, *27*, 721–735. [CrossRef]

89. Hu, P.; Dhelim, S.; Ning, H.; Qiu, T. Survey on fog computing: Architecture, key technologies, applications and open issues. *J. Netw. Comput. Appl.* **2017**, *98*, 27–42. [CrossRef]

90. Naha, R.K.; Garg, S.; Georgakopoulos, D.; Jayaraman, P.P.; Xiang, Y.; Ranjan, R.; Gao, L. Fog computing: Survey of trends, architectures, requirements, and research directions. *IEEE Access* **2018**, *6*, 47980–48009. [CrossRef]

91. Shi, W.; Cao, J.; Zhang, Q.; Li, Y.; Xu, L. Edge Computing: Vision and Challenges. *IEEE Internet Things J.* **2016**, *3*, 637–646. [CrossRef]