Future Generation Computer Systems 91 (2019) 407–415

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Task scheduling techniques in cloud computing: A literature survey

AR. Arunarani [a], D. Manjula [a], Vijayan Sugumaran [b,c,*]

[a] Department of Computer Science and Engineering, Anna University, India
[b] Department of Decision and Information Sciences, Oakland University, Rochester, MI, USA
[c] Center for Data Science and Big Data Analytics, Oakland University, Rochester, MI, USA

HIGHLIGHTS

- Presents a comprehensive survey of task scheduling strategies and the associated metrics suitable for cloud computing environments.
- Discusses the various issues related to scheduling methodologies and the limitations to overcome.
- Distinctive scheduling procedures are studied to discover which characteristics are to be included in a given system and which ones to disregard.
- Literature survey organized based on three different perspectives: methods, applications, and parameter-based measures utilized.
- Future research issues related to cloud computing-based scheduling identified.

ARTICLE INFO

Article history:
Received 23 September 2017
Received in revised form 15 August 2018
Accepted 5 September 2018
Available online 22 September 2018

Keywords:
Cloud computing
Task scheduling
Scheduling algorithms
Resource allocation
Scheduling strategy
Scheduling metrics

ABSTRACT

Cloud computing manages a variety of virtualized resources, which makes scheduling a critical component. In the cloud, a client may utilize several thousand virtualized assets for every task. Consequently, manual scheduling is not a feasible solution. The basic idea behind task scheduling is to slate tasks to minimize time loss and maximize performance. Several research efforts have examined task scheduling in the past. This paper presents a comprehensive survey of task scheduling strategies and the associated metrics suitable for cloud computing environments. It discusses the various issues related to scheduling methodologies and the limitations to overcome. Distinctive scheduling procedures are studied to discover which characteristics are to be included in a given system and which ones to disregard. The literature survey is organized based on three different perspectives: methods, applications, and parameter-based measures utilized. In addition, future research issues related to cloud computing-based scheduling are identified.

1. Introduction

The collection of interconnected computers that consists of more than one united computing resource is known as the Cloud. In recent years, the advancement of cloud computing has helped simulate the quick arrangement of inter-connected data centers that are geographically dispersed for offering high quality and dependable services [1]. These days, cloud computing has turned into an efficient paradigm to offer computational abilities on a "pay-per-utilize" premise [2]. Cloud computing brings the conformity and change in the IT business. With its developing application and promotion, cloud computing offers tremendous open doors, as well as confronts many difficulties in the advancement of traditional IT [3]. Recently, cloud computing has risen as another Internet-based model for empowering clients. It can organize access to a shared pool of configurable assets on-request, which can be immediately provided and discharged with very little administration or cloud provider cooperation [4]. Because of this innovation, many advantages such as improved benefits in the market place with respect to time, cost, stack adjusting, and storage can be realized. With this innovation, all applications can keep running on a virtual platform and every one of the resources is distributed among the virtual machines [5]. Every last application is distinctive and independent.

Some of the parallel applications show a decrease in utilization of CPU resources whenever there is an increase in parallelism. If the jobs are not scheduled correctly, performance reduces because the cloud processes a huge amount of data. Thus, the scheduling mechanism plays a vital role in cloud computing. A scheduling algorithm is utilized to plan the task with greatest evaluated gain or benefit and execute the task. However, computing ability in the distributed system shifts from various resources to the cost of resource utilized. The distributed computing administrative tasks such as stockpiling and data transfer processes are easy to manage and bring down expenses. These tasks are scheduled in view of

* Corresponding author at: Department of Decision and Information Sciences, Oakland University, Rochester, MI, USA.
E-mail address: sugumara@oakland.edu (V. Sugumaran).

https://doi.org/10.1016/j.future.2018.09.014
0167-739X/

client's necessity. Additionally, if the number of clients using the cloud increases, scheduling becomes quite difficult and an appropriate scheduling algorithm needs to be utilized. In the early stage, some of the scheduling algorithms were developed in the context of grid computing and based on their performance many were adapted for distributed computing. In cloud computing, users may utilize hundreds or thousands of virtualized resources and it is impossible for everyone to allocate each task manually. Due to commercialization and virtualization, cloud computing handles the task scheduling complexity in the virtual machine layer. Thus, scheduling plays an important role in cloud computing to assign the resources to each task efficiently and effectively. Nowadays, different types of scheduling mechanisms are available such as cloud service scheduling, heuristics scheduling, workflow scheduling, static scheduling and dynamic scheduling. In the cloud, the internal and external requirements of the resources are maintained and the requirements such as bandwidth, storage, resource expenses, and response time may differ for each task. Load balancing, scalability, reliability, performance and dynamic re-allocation of resources to the computing nodes are all the major problems that manifest in task scheduling. Hence, an efficient scheduling algorithm is needed for task scheduling in the cloud computing environment.

In order to develop effective scheduling algorithms, we need to clearly understand the various problems associated with different scheduling methodologies and the limitations to overcome. Thus, the objective of this paper is to present a comprehensive survey of task scheduling strategies and the associated metrics suitable for cloud computing environments. As part of this work, we have studied the different distinctive scheduling procedures to identify which characteristics are to be included in a given system and which ones to disregard. The literature survey is organized based on different perspectives.

The remainder of this article is organized as follows. Section 2 presents a detailed survey of the different scheduling approaches that have been reported in the literature in the last decade. Section 3 organizes the existing task scheduling works based on technique, applications, and parameter-based measures utilized. Section 4 provides conclusion and future work.

2. Survey of scheduling in cloud computing

All articles that had the word "scheduling" in the title or keyword, published from January 2005 to March 2018, were first selected from scientific journals including IEEE, Elsevier, Springer and other international journals. A huge number of studies have been devoted to machine learning and other techniques to work through problems in cloud computing. This section surveys and categorizes the various task scheduling techniques. They can be subdivided into ten broad categories: QoS-based task scheduling, Ant Colony Optimization Algorithm-based scheduling, PSO-based task scheduling, Genetic Algorithm-based task scheduling, Multiprocessor-based scheduling, Fuzzy-based scheduling, Clustering-based task scheduling, Deadline-constrained scheduling, Cost-based scheduling and other scheduling-based approaches. Each of these categories is briefly discussed in the following subsections.

2.1. QoS parameter based task scheduling algorithm

Distributed computing is to empower access to a common pool of assets through an Internet service. With the number of cloud clients is duplicating each day, task scheduling turns into an imperative issue to manage. Task scheduling is to dispense the accessible assets without abusing or over-burdening a specific framework to the incoming jobs. Proficient task scheduling upgrades the execution of the cloud and gives better support of the cloud clients. The scheduling done is assessed in view of various criteria known as the Quality of Service parameters. There are many algorithms for task scheduling enhancing various QoS parameters. One of the first such efforts was that of He et al. [6], who proposed the QoS-guided Min–Min heuristic for grid task scheduling. This algorithm was based on a general adaptive scheduling heuristic that comprises QoS guidance. They assessed the algorithm inside a simulated grid environment, but several issues remain open, including the fact that they addressed only one-dimensional QoS issues. To improve resource utilization, they incorporated distributed QoS-constraint-based task scheduling in the cloud from the work of Patel and Bhavsar [7], and Li et al. [8]. Initially, they set up the QoS-differentiated system model for sharing resources with different QoS-constrained users in the cloud computing system. Next, they introduced an optimizing chord algorithm to schedule tasks submitted by users with lower QoS constraints. This work has not considered the influence of networks and interaction with storage resources. Similarly, Dubey and Agrawal [9] discussed QoS-driven task scheduling in cloud computing. They made use of the fixed-priority scheduling algorithms, i.e. Rate Monotonic and Deadline Monotonic, for task scheduling as priority-based strategies. On the other hand, Wu et al. [10] developed a QoS-driven task scheduling algorithm in cloud computing. This method takes the most time to complete the scheduling. In [11], Albodour et al., have discussed business grid operations relative to its QoS and rescheduling capabilities. They present the concept of Business Grid QoS (BGQoS) and investigate the behavior and performance of different operations and components within BGQoS. They provide a comparison between the different operations and their effect on the full model. Moreover, Ali et al. [12] have explained a Grouped tasks scheduling algorithm based on QoS in cloud computing network. This algorithm was distributed tasks into five categories; each category has tasked with similar attributes (user type, task type, task size, and task latency) Similarly, Faruk and Sivakumar [13] presented a multi-layer QoS-based task scheduling algorithm for cloud environments. The experimentation was based on the CloudSim algorithm which offers good presentation and load balancing from a QoS standpoint, in terms of both priority and completion time. Moreover, Hayyolalam and Kazem [14] have explained a QoS-aware service composition and selection in a cloud environment. Here, the author review lot of published paper. The limitation and merits of each paper are analyzed.

2.2. Ant colony optimization algorithms based scheduling

A good task scheduler should adjust its scheduling technique to the changing condition and the kinds of task. Consequently, a dynamic task scheduling calculation, for example, Ant Colony Optimization (ACO), is fitting for clouds. ACO calculation is an arbitrary search algorithm. This algorithm utilizes a positive criticism system and mimics the conduct of genuine ant colonies in nature to scan for food and to interface with each other by pheromone laid on ways voyaged. Numerous scientists utilized ACO to take care of NP-difficult issues, for example, voyaging sales representative issue, graph coloring issue, vehicle routing issue and scheduling issue. Earlier, a lot of researchers used this algorithm to explain the scheduling approach. Mathiyalagan et al. [15] used an enhanced ant colony algorithm to describe grid scheduling. To develop the competence of the system, they further brought in an adapted ant colony optimization algorithm for programming in grid computing, as exemplified in the work of Mathiyalagan et al. [16]. In addition, Maruthanayagam and Uma Rani [17] explained scheduling in grid computing using the enhanced Ant Colony System based on the RASA algorithm. The distribution of resources to a huge number of jobs in a grid computing environment is more complex than in network computational environments. Similarly, Delavar et al. [18] discussed task scheduling in the grid environment with a metaheuristic ant colony optimization method in regard to cost and time parameters from the perspective of Quality of Service.

Srikanth et al. [19] introduced task scheduling by means of ant colony optimization. ACO, a bio-inspired computing paradigm, was employed for generating the program, but the average waiting time between tasks was long. Similarly, Khan and Sharma [20] introduced effective load balancing in cloud computing using the ACO. They contrasted the ACO algorithm with others to summarize the most recent algorithms for use in the cloud. However, these algorithms are, usually, inflexible and cannot match the dynamic changes to the attributes during the execution. In addition, the adapted ACO algorithm was proposed to program tasks, as propounded by Tawfeek et al. [21]. In [22], Niazmand et al. discussed workflow scheduling in grid computing by means of an improved algorithm for ant colony optimization. The JSWA algorithm was appropriately employed to calculate parameters such as dependability, costs, requests, acknowledgment time, and bandwidth. Elayaraja and Dhanasekar [23] presented workflow scheduling using heuristics-based ant colony optimization. The Ant Colony System (ACS) is an algorithm in the ACO, based on the performance of ants. This paper overcomes the difficulties of ACO-based scheduling even though this method is computationally complex and its total completion time is also protracted. On the other hand, Khambre et al. [24] created the modified pheromone update rule to execute the ant colony optimization algorithm for the workflow scheduling algorithm problem in grids. The QoS parameters considered in the simulation were time, cost, and budget, and the optimization was based on user-defined QoS parameters. Even though this particular ACO algorithm reduced problems with the overhead, it culminated in a poor response time.

2.3. PSO based task scheduling

Numerous meta-heuristic calculations have been proposed, for example, PSO algorithm that is suitable for dynamic task scheduling incorporate. Then again, Particle Swarm Optimization (PSO) has turned out to be popular due to its effortlessness and its adequacy in an expansive scope of utilization. This algorithm works based on the behavior of birds. A portion of the applications have utilized PSO to take care of NP-Hard issues like Scheduling issue, and the resource allocation issue. Some of the works related to PSO based task scheduling is explained below. Pandey et al. [25] offered a scheduling heuristic method based on particle swarm optimization (PSO) to reduce the total cost of execution. This study contrasted the PSO and 'Best Resource Selection' (BRS) algorithms and the results illustrate that the PSO is an improvement, since it attains three times as much cost savings as compared to the BRS, alongside an enhanced distribution of workload with the resources at hand. As a result, the transfer of data from one compute node to another takes longer, incurring higher transmission and storage costs. Feng et al. [26] briefly touched on the crisis of resource allocation. A particle swarm optimization algorithm was planned to tackle this problem by bringing in the Pareto dominance theory. This theory investigates optimal schedulers in the multi-objective optimizing matter of resources, based on the total task executing time, resource reservation, and QoS of each task. However, this work only used simple tasks, with no general convergence theory applicable to task scheduling problems.

Scheduling is a major issue in cloud computing and a scheduling a mechanism is necessary to enhance performance and utilize resources effectively. Scheduling involves allocating certain jobs to particular resources at particular times. To improve the PSO algorithm for task scheduling, Juan et al. [27] advanced an improved PSO-based task scheduling algorithm for cloud storage systems to overcome difficulties. They presented an enhanced PSO-based algorithm by defining a cost vector and restricting the initialization solution and the solution search space in the Exist Solution Space. They generated Cost Vector model which was used to measure the cost of the scheduling scheme and also solution was initialized based on the input task and QoS parameters. This method was effective but it has a lot of complexity. Guo et al. [28] proposed task scheduling optimization in cloud computing based on a heuristic algorithm. This work presents an optimization policy that only optimizes efficiency, not energy and service-level agreements.

To overcome the difficulties present in PSO-based scheduling, Umale et al. [29] suggested a modified discrete particle swarm optimization to optimize workflow application programs in the cloud computing environment. As a result, some parts of the search space may be unreachable. Similarly, a study by Huang [30] implemented particle swarm optimization (PSO) for workflow scheduling in cloud computing. Most of the studies in this area spotlighted a single objective. This study clearly explained a tunable fitness function for the PSO algorithm, based on which a workflow program may be chosen at a minimal cost or a minimal makespan (completion time), or any level in between. A heuristic was developed to address bottleneck problems and reach a smaller makespan. Gomathi and Krishnasamy [31] developed a task scheduling algorithm based on hybrid particle swarm optimization, which enhances particle swarm optimization, condenses the average operation time, raises the utilization ratio of resources, and supplies suitable resources to the user's task competently. But it does not work on large-scale optimization. Kaur and Sharma [32] used the Hybrid Improved Particle Swarm Optimization for resource utilization. This HIPSO algorithm evades sinking into the local optima and raises the convergence speed of the PSO as well. Moreover, Alkayal et al. [33] have explained a multi-object based task scheduling using particle swarm optimization algorithm based on a new ranking strategy. The main insight of this algorithm was that the tasks were scheduled to the virtual machines to minimize waiting time and maximize system throughput. To overcome the difficulties Dordaie and JafariNavimipour [34] have explained a hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments. This method was scheduled properly. Even though, the hybrid algorithm takes maximum time to complete the task. Similarly, Verma and Kaushal [35] have explained hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling.

2.4. Genetic algorithm based task scheduling

Several methods have been proposed to handle the issue of GA-based task scheduling. Of these, some of the research is analyzed here. Gao et al. [36] introduced job shop scheduling using a multi-objective function. They analyzed the flexible job shop scheduling problem (FJSP) to reduce the execution time and execution cost of the scheduling algorithm. S. Kaur and Verma [37] discussed the scheduling algorithm. To overcome the difficulties of scheduling, Kumar and Verma [38] improved scheduling performance with Min–Min and Max–Min methods combined with a standard genetic algorithm to generate an enhanced GA. Jang et al. [39] elaborated on task scheduling using a genetic algorithm. In [40], Cheng explained cloud service workflow scheduling and optimization schema using hierarchical cloud service workflow scheduling. Cloud workflow tasks parallel the split-, syntax- and semantic-based cloud workflow task matching algorithm, and multiple QoS constraint-based cloud workflow scheduling and optimization. It offers an experimental analysis of the algorithm's competence. In addition, to achieve the maximum utilization of resources, they incorporated research by Kaleeswaran et al. [41] on the dynamic scheduling of data by means of a genetic algorithm in cloud computing. Savitha and Reddy [42] reviewed task scheduling using a genetic algorithm. In addition, they clarified the load balancing strategy for cloud computing by means of the genetic algorithm in Dasgupta et al. [43]. To enhance the overall performance of cloud computing with the deadline constraint, a task scheduling model was launched to decrease the system's power consumption and maximize the profits of service providers in the multi-objective genetic algorithm (MO-GA) of Liu et al. [44]. This system was planned by means of encoding rules, crossover operators, selection operators and the technique of sorting Pareto solutions. Based on the open source cloud computing simulation platform CloudSim, and in contrast to the scheduling algorithms presented, the results illustrate that the algorithm offers an improved solution and a balanced multiple-object performance. Similarly, Xu et al. [45] propositioned a genetic algorithm for task scheduling on heterogeneous computing systems by means of multiple priority queues (MPQGA). Moreover, Keshanchi et al. [46] have explained an improved genetic algorithm for task scheduling in the cloud environments using the priority queues. In this method, Elitism technique with unusual selections was adopted to evade premature convergence and the Statistical analyzes on the different randomly generated graphs were done. Additionally, Shishido et al. [47] have explained workflow scheduling algorithm using a genetic algorithm. Here, a security and cost-aware workflow scheduling algorithm were selected to evaluate the performance of the metaheuristics.

2.5. Multi-processor based scheduling

For the purpose of this research, the earlier work associated with the scheduling used in multiprocessors was studied. A few current studies are briefly discussed in this section. Kwok and Ahmad [48] devised optimal algorithms for the static scheduling of task graphs with random parameters for multiple homogeneous processors. The initial algorithm was based on the image search technique and employed a computationally-competent cost function for guiding the search with little difficulty. With this method, the running time of the execution process increases. To tackle this issue, Mohamed and Awadalla [49] proposed two approaches, a Modified List Scheduling Heuristic (MLSH) and a hybrid approach composed of a Genetic Algorithm (GA) and the MLSH for task scheduling in multi-processor systems. Additionally, the authors devised three different representations for the chromosomes of the genetic algorithm: a task list (TL), a processor list (PL), and a combination of both (TLPLC). The disadvantage of this work is that multiprocessor task scheduling takes the longest execution time. Dhingra et al. [50] introduced multiprocessor task scheduling using a genetic algorithm. The results of the GA parameters on minimizing the bi-criteria fitness function and parameter setting was completed by the central composite design (CCD) approach of the response surface methodology (RSM) of the Design of Experiments. Equipped with the minimum speed, each task has some prerequisite constraints that need to be satisfied. It means that a given task cannot be executed until the prior tasks are first satisfied.

2.6. Fuzzy based scheduling

Fuzzy-based scheduling algorithms are briefly reviewed in this section. Fahmy [51] explained scheduling non-periodic jobs using a fuzzy algorithm in real-time systems. The algorithm presumes that there is a heavily loaded machine with a single processor distributed by multiple users. The plan was to employ a fuzzy logic algorithm to examine the priority of the job to be implemented first. A second fuzzy algorithm was employed to acclimatize the priorities of other jobs in waiting, in case a novel job appeared and set deadlines for these jobs. This fuzzy logic load scheduling algorithm was employed inside a multi-objective algorithm to reduce the average delay, the number of overdue jobs, and the throughput times of the jobs. The purposes were to decrease job total throughput time and enhance user satisfaction. On the other hand, the virtualization is disputed, chiefly owing to resource management and task scheduling prevailing over the competent dynamic task scheduling in virtualized data centers with fuzzy prediction, as explained by Kong et al. [52]. By employing type-I and type-II fuzzy logic systems, an elegant fuzzy prediction method was specified to model the uncertain workload and vague accessibility of virtualized server nodes. An on-line dynamic task scheduling algorithm named SALAF was brought in and assessed. They additionally integrated robotic flexible assembly cells into the system, based on the fuzzy approach of Abda et al. [53], to develop a competent rule for scheduling. They employed a fuzzy sequencing rule (FSR), which was erected by combining dissimilar input variables: the processing time, due date, batch size and number of assembly stations necessary, using the fuzzy logic (FL) technique.

Suer et al. [54] propositioned the evaluation of feedback among multiple scheduler profiles in fuzzy genetic scheduling. This study examines in detail earlier studies performed on multiple scheduler profiles in fuzzy genetic scheduling. Multiple schedulers were arranged in individual fuzzy membership bounds which affect the assessment of multi-objective problems of single-machine scheduling. A novel software application facilitates feedback among schedulers by seeding an individual scheduler's population with the best chromosomes from another scheduler's population. Mehranzadeh and Hashemi [55] utilized fuzzy logic for task scheduling. Their study offers and assesses a new scheduling algorithm that is competent at scheduling virtual machines in data centers. The simulation effect demonstrates the efficiency of the algorithm by contrasting it with two scheduling techniques, First Fit (FCFS) and Round Robin (RR). The results reveal that the scheduling algorithm is successful in the cloud. This algorithm affects exterior priorities when scheduling multiple jobs, and rule generation is a very difficult task in fuzzy logic since it impacts the computation time. Moreover, Priya and Babu [56] have explained a Moving average fuzzy resource scheduling for virtualized cloud data services. A fuzzy control theory was designed for system accessibility between user cloud requirements and cloud users resources.

2.7. Clustering based task scheduling

Qin and Jiang [57] expounded a heuristic dynamic scheduling scheme for parallel real-time jobs, implemented by means of a heterogeneous cluster. A parallel real-time job modeled by directed acyclic graphs disembarks at a heterogeneous cluster, following a Poisson process. A job is said to be possible if all its tasks meet their relevant deadlines. The scheduling algorithm takes dependability measures into account, thus improving the dependability of heterogeneous clusters at no extra hardware cost. Papazachos and Karatza [58] proposed gang scheduling, based on two clustering systems. Gang scheduling is a general task scheduling policy for parallel and distributed systems which unites elements of space-sharing and time-sharing. They offer a migration approach which decreases the fragmentation in the schedule caused by gang programmed jobs. A distributed system contains two homogeneous clusters, reproduced to assess the presentation for different workloads. They discussed the impact of changeability on the service time of parallel tasks. When using the clustering algorithm for task scheduling, it exhibits an inability to make corrections once the splitting/merging decision is made. Zhang et al. [59] developed a parallel task scheduling algorithm based on fuzzy clustering in the cloud computing environment. Parallel task scheduling is a major problem in the field of cloud computing. Their research focused on parallel scheduling, with special reference to the high-presentation computing necessary for massive data processing, especially in the oil and seismic exploration sectors. The drawback of their clustering algorithm is the lack of interpretability concerning cluster descriptors. Similarly, Abraham et al. [60] have presented a group-based parallel multi-scheduler for grid computing. It effectively exploits the benefits of multicore systems for Grid by splitting the jobs and machines into paired groups and independently multi-scheduling jobs in parallel to the groups. The Priority method splits jobs into four priority groups based on job attributes and uses two methods (SimTog and EvenDist) to group machines. Then the scheduling is carried out using the Min–Min algorithm within the discrete group pairs. Dandhwani and Vekariya [61] have explained a multi-objective task scheduling algorithm using K-means for cloud computing. Using this method, we have to predict the K-Value. Yokoyama et al. [62] have discussed an affinity aware scheduling model of cluster nodes in private clouds. It mainly focuses on system throughput. Based on co-allocated job interactions, it selects the host with better throughput.

2.8. Deadline constrained scheduling

In this section, deadline-constrained task scheduling is explored. Abrishamiand and Naghibzadeh [63] introduced deadline-constrained workflow scheduling in Software as a Service in the cloud. Abrishami et al. [64] presented deadline-constrained workflow scheduling algorithms for Infrastructure as a Service. They acclimatized the PCP algorithm for the cloud environment and suggested two workflow scheduling algorithms: a one-phase algorithm called the IaaS Cloud Partial Critical Paths (IC-PCP), and a two-phase algorithm called the IaaS Cloud Partial Critical Paths with Deadline Distribution (IC-PCPD2). Both algorithms have a polynomial time difficulty which makes them appropriate options for scheduling large workflows. Workflow execution is, however, susceptible to delays if one or more of the virtual machines (VMs) fail during task execution.

2.9. Cost-based scheduling

This section looks at cost-based scheduling algorithms. Su et al. [65] explained cost-efficient task scheduling for executing large programs in the cloud. They employed a cost-efficient task-scheduling algorithm by means of two heuristic strategies. The first approach dynamically maps tasks to the most cost-efficient VMs based on the idea of Pareto dominance. The second approach, a complement to the first strategy, decreases the monetary costs of non-critical tasks. In addition, cloud service providers, leasing resources from cloud vendors under the pay-per-use service model, would want to reduce rental costs while meeting users' computing needs.

2.10. Other approaches for scheduling

In recent years, other methods have been proposed for scheduling as well. Son et al. [66] have elaborated on extracting the workflow critical path from the extended well-formed workflow plan. First, they explained the workflow model with a set of workflow control erects that offer adequate power to drive the models of most of today's business processes. Next, they devised a systematic method of recognizing significant paths for a specified workflow schema. In addition, Senkul and Toroslu [67] developed an appropriate architecture to identify and schedule workflows under resource allocation constraints while operating under temporal and causality constraints.

Wang et al. [68] proposed the Cloud-DLS, a dynamic trusted programming model for cloud computing. The cloud is quickly turning into an imperative platform for scientific applications. In the cloud environment that operates with countless nodes, resources are inevitably unreliable. Task execution and scheduling can make a discernible difference in such a set-up. This study is motivated by the Bayesian cognitive model and references the trust relationship models of sociology. The Bayesian method-based cognitive trust model works alongside a trust dynamic-level scheduling algorithm called the Cloud-DLS by integrating the presented DLS algorithm.

Babu and Krishna [69] presented honeybee behavior-inspired load balancing (HBB-LB) to attain well-balanced loads across virtual machines for maximizing the throughput. In [70], Lee et al. discussed the competent scheduling algorithm for component-based networks, which is capable of rapidly quantifying the Quality of Service attainable by each alternative composition of resources in a grid computing environment. Competence is a crucial driver that proficiently utilizes resources and promotes the effective economy. The presentation of the network is a function of resource assignment and resource allocation: the former allocates components to access machines and the latter assigns the resources of each machine to the residing components. Though related problems can be found in the multiprocessor scheduling literature, their problem is dissimilar particularly because the components in their networks process multiple tasks in parallel with their successor or predecessor components. The planned algorithm is uncomplicated but efficient as it integrates components in a network that can be regarded as independent under a certain resource allocation policy.

2.11. Summary of survey

Task scheduling plays a crucial part in managing and sharing cloud resources with different cloud users. Therefore, task scheduling is a major research topic in the area of cloud computing. In this survey paper, we have analyzed the concept of task scheduling published in the literature between January 2003 and March 2018. We selected articles from scientific journals from publishers such as IEEE, Elsevier, Springer, and other international journals. During this period, a huge number of studies have been conducted on scheduling and different techniques to solve problems related to scheduling. Here, we have analyzed more than sixty research works which have different applications such as task scheduling, resource allocation, and load balancing. It is found that a variety of scheduling algorithms work on different scheduling criteria and that all algorithms are efficient in one way or another. Each method has advantages and disadvantages. Therefore, there is every possibility of continuing and enhancing previously completed work in this field. Each algorithm has some limitations such as maximum scheduling time, overloaded, computation complexity and delay. Moreover, most of the research works have only used a small number of tasks as well as one dimensional tasks. In addition, some of the research works only consider a single objective function based task scheduling. Compared to multi-objectives, a single objective cannot provide maximum results. Moreover, the limitation of Min–Min algorithm [6] is that it chooses smaller tasks first which makes use of resources with high computational power unnecessarily. As a result, the schedule produced by Min–Min is not optimal when number of smaller tasks exceeds the large ones. Similarly, Max–min algorithm schedules larger tasks first. But in some cases, the makespan may increase due to the execution of larger tasks first. The waiting time of smaller tasks is also increased in Max–min [38]. To overcome the challenges present in current scheduling algorithms, there is a great need for developing novel approaches.

Table 1
Overall analysis of Survey 2005–2018.

Technique	2005–2008	2009–2012	2013–2016	2017–2018
QoS	[6],	[7]	[9,10,13,11]	[12,14]
Ant	–	[17–19]	[22–24]	–
PSO	–	[25,27,26,28]	[29–33]	[34,35]
GA	[36]	[49,37,39,40,38]	[41–45,50]	[46,47]
ACO	–	[15,16]	[20,21]	–
Fuzzy	–	[51–54]	[55,70]	[56]
Clustering	[57]	[58]	[59,60]	[61]
Other approaches	[48,66,65,69]	[62,67,8]	[63,68,64]	–

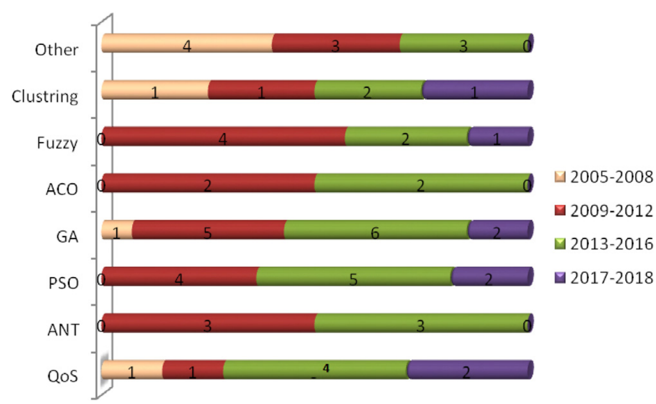

Fig. 1. Summary of the survey.

3. Categorization and discussion

In this section, we categorize all the articles considered in the survey, based on three different criteria: technique, application, and parameter-based measures. Each category is briefly described in the following sub-sections.

3.1. Categorization based on techniques

We have categorized the existing research works in accordance with the technique used. This analysis will be very much helpful in understanding which algorithms were mainly utilized in the earlier days and how newer algorithms were further added based on additional requirements. Table 1 shows when each of the techniques were introduced and the research works that utilized them in different years. Here, we split the existing literature on task scheduling into four different time periods as follows: 2005–2008, 2009–2012, 2013–2016 and 2017–2018. The listing of the specific research works utilizing different techniques in each time period is shown in Table 1. For this survey, totally sixty-three papers were considered, and each paper has utilized a different algorithm for task scheduling. As seen from Table 1, during the 2005–2008 period, seven papers have been published. Similarly, during the period 2009–2012 23 papers, 2013–2016 27 papers, and 2017–2018 eight papers have been published, respectively. From Fig. 1, it is evident that during the years 2009–2012 and 2013–2016, there has been a flurry of research activity and publication in task scheduling using different techniques. In particular, the Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) techniques have been heavily used by different researchers in developing task scheduling approaches for cloud computing.

3.2. Categorization based on application

In this categorization, the scheduling application that has been targeted in each of the 65 research articles that we considered is identified. Table 2 shows the scheduling application, type of scheduling, and the environment for all the 65 papers. The application categories are job scheduling, workflow scheduling, task scheduling, and others. Similarly, the types of scheduling are static scheduling, dynamic scheduling. The environment types are cloud environment and grid environment. Of the 65 papers, 16 focused on job scheduling, 10 on workflow scheduling, 35 on task scheduling, and three papers presented resource allocation and load balancing. Most of the articles used dynamic scheduling to organize tasks, and only five were based on the static scheduling approach. In addition, most of the articles focused on utilizing the cloud environment, with only 10 using the grid environment for scheduling.

3.3. Categorization based on parameter measure

This section explains the parameters used to measure effectiveness of scheduling. The existing works have used different kinds of measures such as makespan, profit, completion time, cost, waiting time, bandwidth, budget efficiency, etc. Table 3 shows the various parameter measures utilized in the literature that we surveyed. As seen from Table 3, completion time is the most frequently used metric for evaluating the effectiveness of the scheduling algorithms that were developed during 2005–2018 time period. Similarly, makespan and cost are the second and third most frequently used parameters during the same time period, respectively.

This survey is intended to be used chiefly for developing novel ideas for future research. From the analysis, it is clear that most of the works have some limitations such as maximum cost, difficult calculation for finding the optimal solution, maximum completion time, less profit and makespan. To overcome these limitations, a new generation of algorithms needs to be developed. The tables shown above summarize most of the existing task scheduling algorithms used in the cloud computing environment. The main objective of scheduling is to minimize resource starvation and to ensure fairness amongst the parties utilizing the resources. Scheduling deals with the problem of deciding which of the outstanding requests is to be allocated resources. This survey can be used to highlight precisely where new algorithms are needed for an enhanced cloud scheduling performance. Varied techniques including heuristics, mathematical operations, and machine learning models as well as evolutionary algorithms – such as the genetic algorithm, ant colony optimization algorithm, cuckoo search algorithm, and swarm intelligence algorithm – can be used to develop hybrid algorithms that can perform more effectively compared to existing approaches. Consequently, future research on task scheduling in cloud computing must concentrate on designing better scheduling techniques based multi-objective function [71] and add some more parameters to improve the performance of the cloud scheduling system.

4. Conclusion and future research

Cloud computing is user-oriented technology wherein users get to choose from hundreds of thousands of virtualized resources for each task. Here, scheduling is considered a major factor for task execution in the cloud environment. In this survey article, we have analyzed various scheduling algorithms and tabulated different parameters used under the cloud and grid environments. In all, 65 articles associated with scheduling, from 2003 to 2018, have been examined. The articles are categorized, year-wise, into three different categories. They have been studied and their limitations and time complexity highlighted. It is, therefore, necessary to improve the availability and reliability of task scheduling within cloud computing. Finally, specific research issues have been identified that need to be addressed in a comprehensive manner. This

Table 2
Categorization based on application from 2005 to 2018.

Articles from 2005 to 2018								
Reference	Job scheduling	Workflow scheduling	Task scheduling	Other	Static scheduling	Dynamic scheduling	Cloud environment	GRID environment
[6]			✓			✓		✓
[7]	✓					✓		✓
[8]			✓			✓	✓	
[9]			✓				✓	
[10]			✓				✓	
[11]	✓				✓	✓		✓
[12]			✓			✓		
[13]			✓			✓		
[14]			✓			✓		
[15]			✓		✓			✓
[16]	✓					✓		✓
[17]	✓					✓		✓
[18]			✓			✓		✓
[19]			✓				✓	
[20]	✓						✓	
[21]			✓			✓	✓	
[22]		✓				✓		✓
[23]		✓						✓
[24]		✓			✓	✓		✓
[25]		✓				✓	✓	
[26]				Resource Allocation			✓	
[27]			✓				✓	
[28]			✓			✓	✓	
[29]	✓						✓	
[30]		✓					✓	
[31]			✓			✓	✓	
[32]			✓			✓	✓	
[33]			✓			✓	✓	
[34]			✓		✓	✓	✓	
[35]	✓					✓	✓	
[36]	✓					✓		
[37]			✓				✓	
[38]			✓				✓	
[39]		✓	✓				✓	
[40]							✓	
[41]			✓			✓	✓	
[42]			✓		✓		✓	
[43]	✓					✓	✓	
[44]	✓					✓	✓	
[45]			✓			✓	✓	
[46]	✓					✓	✓	
[47]			✓			✓	✓	
[48]			✓		✓		✓	
[49]			✓				✓	
[50]			✓				✓	
[51]	✓							
[52]			✓				✓	
[53]				✓		✓		
[54]	✓							
[55]	✓						✓	
[56]			✓			✓	✓	
[57]	✓					✓	✓	
[58]				✓		✓	✓	
[59]			✓			✓	✓	
[60]			✓			✓		✓
[61]			✓			✓	✓	
[62]		✓				✓	✓	
[63]		✓				✓	✓	
[64]			✓			✓	✓	
[65]		✓			✓	✓		
[66]		✓				✓		✓
[67]	✓			✓			✓	
[68]						✓	✓	
[69]			✓		✓			✓
[70]			✓			✓	✓	

will provide future avenues and encourage researchers to conduct further research along these directions.

Task scheduling is one of the most important problems in cloud computing, so there is always an opportunity for the modification of previously completed work. Researchers have conducted their work at a particular point in time, limited by the constraints in terms of knowledge, space and time. With the passage of time, however, their work has been improved on as a matter of course by other researchers. Miscellaneous techniques have been considered during scheduling and numerous constraints applied, but given the vastness of cloud computing, researchers have been unable to capture all of its aspects simultaneously. They have observed

Table 3
Categorization based on parameter measure.

Measures	2005–2008	2009–2012	2013–2016	2017–2018
Makespan	[1]	[14–16,23,29,26]	[35,38,39,56,47,49]	[62]
Profit			[43,55,33]	
Completion time	[1,5–7]	[9,10,13,15,5,17,20–22,25,29]	[33,45,50,54,56,53,48,40,51,52]	[65]
Cost	[3,5]	[8,17,20,22,23,27]	[46,36,51,52]	[61,62]
Waiting time	[4]	[12,18]		[64]
Other	[2,31]	[19,27,30]	[32,41,57,11,60]	[59,63]

that certain algorithms are open to modification, and identified the parts needing further modifications. Future research should focus on how to effectively combine task scheduling and virtual machine consolidation strategies to further enhance the effectiveness of scheduling. In order to improve popular and classic scheduling techniques in cloud computing, new methods need to be developed which include economic models and heuristic algorithms along with algorithms inspired by nature [72–74]. By combining different approaches and considering input parameters such as running costs and deadlines, it is possible to provide a powerful approach for scheduling tasks in a cloud computing environment. Further, future work should also pursue single objective- and multi-objective-based task scheduling using different hybridization of existing algorithms.

Acknowledgment

The research work by Vijayan Sugumaran has been supported by the 2018 School of Business Administration Spring/Summer Research Fellowship from Oakland University, United States.

References

[1] W. Kaiyue, P. Lu, Z. Zhu, Distributed online scheduling and routing of multicast-oriented tasks for fitness function-driven cloud computing, IEEE Commun. Lett. 20 (4) (2016) 684–687.
[2] X. Zhu, C. Chen, L.T. Yang, Y. Xiang, ANGEL: Agent-based scheduling for real-time tasks in virtualized clouds, IEEE Trans. Comput. 64 (12) (2015) 3389–3403.
[3] C. Cheng, J. Li, Y. Wang, An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing, Tsinghua Sci. Technol. 20 (1) (2015) 28–39.
[4] B. Keshanchi, A. Souri, N.J. Navimipour, An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing, J. Syst. Softw. 124 (2017) 1–21.
[5] G. Lovász, F. Niedermeier, H.D. Meer, Performance tradeoffs of energy-aware virtual machine consolidation, Cluster Comput. 16 (3) (2013) 481–496.
[6] X. He, X. Sun, G. von Laszewski, QoS guided Min-Min heuristic for grid task scheduling, J. Comput. Sci. Technol. 18 (4) (2003) 442–451.
[7] S. Patel, M. Bhavsar, QoS based user driven scheduler for grid environment, Adv. Comput.: Int. J. (ACIJ) 2 (1) (2011).
[8] B. Li, A.M. Song, J. Song, A distributed QoS-constraint task scheduling scheme in cloud computing environment: Model and algorithm, Adv. Inf. Sci. Serv. Sci. 4 (5) (2012).
[9] S. Dubey, S. Agrawal, QoS driven task scheduling in cloud computing, Int. J. Comput. Appl. Technol. Res. 2 (5) (2013) 595–600.
[10] X. Wu, M. Deng, R. Zhang, B. Zeng, S. Zhou, A task scheduling algorithm based on Qos -driven in cloud computing, Proc. Comput. Sci. 17 (2013) 1162–1169.
[11] R. Albodour, A. James, N. Yaacob, QoS within business grid quality of service (BGQoS), Future Gener. Comput. Syst. 50 (2015) 22–37.
[12] H.E.D. Ali, I.A. Saroit, A.M. Kotb, Grouped tasks scheduling algorithm based on QoS in cloud computing network, Egyptian Inform. J. 18 (1) (2017) 11–19.
[13] M.N. Faruk, D. Sivakumar, Multi-layer QOS based task scheduling algorithm for cloud environments, Int. J. Adv. Comput. Technol. (2014).
[14] V. Hayyolalam, A.A. Kazem, A systematic literature review on QoS-aware service composition and selection in cloud environment, J. Netw. Comput. Appl. 110 (2018) 52–74.
[15] P. Mathiyalagan, U.R. Dhepthie, S.N. Sivanandam, Grid scheduling using enhanced ant colony algorithm, ICTACT J. Soft Comput. 2 (2010).
[16] P. Mathiyalagan, S. Suriya, S.N. Sivanandam, Modified ant colony algorithm for grid scheduling, Int. J. Comput. Sci. Eng. 2 (2) (2010) 132–139.
[17] D. Maruthanayagam, R.U. Rani, Enhanced ant colony system based on RASA algorithm in grid scheduling, Int. J. Comput. Sci. Inf. Technol. 2 (4) (2011) 1659–1674.
[18] A.G. Delavar, J. Bayrampoor, A.R.K. Boroujeni, A. Broumandnia, Task scheduling in grid environment with ant colony method for cost and time, Int. J. Comput. Sci. Eng. Appl. 2 (5) (2012).
[19] U. Srikanth, V.U. Maheswari, P. Shanthi, A. Siromoney, Tasks scheduling using Ant Colony Optimization, J. Comput. Sci. 8 (8) (2012) 1314–1320.
[20] S. khan, N. Sharma, Ant colony optimization for effective load balancing in cloud computing, Int. J. Emerging Trends Tech. Comput. Sci. 2 (6) (2013).
[21] M.A. Tawfeek, A. El-Sisi, A.E. keshk, F.A. Torkey, An ant algorithm for cloud task scheduling, in: International Workshop on Cloud Computing and Information Security, 2013.
[22] E. Niazmand, A.G. Delavar, J. Bayrampoor, A.R.K. Boroujeni, JSWA: An improved algorithm for grid workflow scheduling using ant colony optimization, J. Math. Comput. Sci. 6 (2013) 315–331.
[23] J. Elayaraja, S. Dhanasekar, Workflow scheduling using heuristics based ant colony optimization, Int. J. Innovative Res. Comput. Commun. Eng. 2 (3) (2014).
[24] P.D. Khambre, A. Deshpande, A. Mehta, A. Saini, Modified pheromone update rule to implement ant colony optimization algorithm for workflow scheduling algorithm problem in grids, Int. J. Adv. Res. Comput. Sci. Tech. 2 (2) (2014).
[25] S. Pandey, L. Wu, S.M. Guru, R. Buyya, A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Enviornments. Technical reports, University of Melbourne, 2009.
[26] M. Feng, X. Wang, Y. Zhang, J. Li, Multi-Objective Particle Swarm Optimization for Resource Allocation in Cloud Computing, IEEE, 2012.
[27] W. Juan, L. Fei, C. Aidong, An improved PSO based task scheduling algorithm for cloud storage system, Adv. Inf. Sci. Serv. Sci. 4 (18) (2012).
[28] L. Guo, S. Zhao, S. Shen, C. Jiang, Task scheduling optimization in cloud computing based on heuristic algorithm, J. Netw. 7 (3) (2012).
[29] J.S. Umale, P.A. Chaudhari, Survey on Job Scheduling Algorithms of Cloud Computing, Int. J. Comput. Sci. Manag. Res. (IJCSMR) (ISSN: 2278-733X) (2013).
[30] J. Huang, K. Wu, L.K. Leong, S. Ma, M. Moh, A tunable workflow scheduling algorithm based on particle swarm optimization for cloud computing, Int. J. Soft Comput. Softw. Eng. 3 (3) (2013).
[31] Gomathi, K. Krishnasamy, Task scheduling algorithm based on hybrid particle swarm optimization in cloud computing environment, J. Theoret. Appl. Inform. Tech. 55 (1) (2013).
[32] G. Kaur, E.S. Sharma, Research paper on optimized utilization of resources using PSO and improved particle swarm optimization (IPSO) algorithms in cloud computing, Int. J. Adv. Res. Comput. Sci. Tech. 2 (2) (2014).
[33] E.S. Alkayal, N.R. Jennings, M.F. Abulkhair, Efficient task scheduling multi-objective particle swarm optimization in cloud computing, in: IEEE 41st Conference on Local Computer Networks Workshops, 2016.
[34] N. Dordaie, N. JafariNavimipour, A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments, J. ICT Exp. (2017).
[35] A. Verma, S. Kaushal, A hybrid multi-objective Particle Swarm Optimization for scientific workflow scheduling, J. parallel computing. 62 (2017) 1–16.
[36] J. Gao, M. Gen, L. Sun, X. Zhao, A hybrid of genetic algorithm and bottleneck shifting for multi-objective flexible job shop scheduling problems, Comput. Ind. Eng.. 53 (1) (2007) 149–162.
[37] S. Kaur, A. Verma, An efficient approach to genetic algorithm for task scheduling in cloud computing environment, Int. J. Inf. Tech. Comput. Sci. 10 (2012) 74–79.
[38] P. Kumar, A. Verma, Independent task scheduling in cloud computing by improved genetic algorithm, Int. J. Adv. Res. Comput. Sci. Softw. Eng. 2 (5) (2012).
[39] S.H. Jang, T.Y. Kim, J.K. Kim, J.S. Lee, The Study of Genetic Algorithm-based Task Scheduling for Cloud Computing, Int. J. Control Autom. 5 (4) (2012).
[40] B. Cheng, Hierarchical Cloud Service Workflow Scheduling Optimization Schema Using Heuristic Generic Algorithm, State Key Laboratory of Networking and Switching Technology., 2012.
[41] A. Kaleeswaran, V. Ramasamy, P. Vivekanandan, Dynamic scheduling of data using genetic algorithm in cloud computing, Int. J. Adv. Eng. Tech. 5 (2) (2013) 327–334.
[42] P. Savitha, J.G. Reddy, A review work on task scheduling in cloud computing using genetic algorithm, Int. J. Sci. Tech. Res. 2 (8) (2013).
[43] K. Dasgupta, B. Mandal, P. Dutta, J.K. Mondal, S. Dam, A genetic algorithm (GA) based load balancing strategy for cloud computing, Proc. Tech. 10 (2013) 340–347.

[44] J. Liu, X.G. Luo, X.M. Zhang, F. Zhang, B.N. Li, Job scheduling model for cloud computing based on multi- objective genetic algorithm, IJCSI Int. J. Comput. Sci. Issues 10 (1:3) (2013).
[45] Y. Xu, K. Li, J. Hu, K. Li, A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues, Inf. Sci. 270 (2014) 255–287.
[46] B. Keshanchi, A. Souri, N. JafariNavimipour, An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: Formal verification. Simulation, and statistical testing, J. Syst. Softw. 124 (2017) 1–21.
[47] H.Y. Shishido, J.C. Estrella, C.F. MottaToledo, M.S. Arantes, Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds, Comput. Electr. Eng. (2017).
[48] Y.K. Kwok, I. Ahmad, On multiprocessor task scheduling using efficient state space search approaches, J. Parallel Distrib. Comput. 65 (12) (2005) 1515–1532.
[49] M.R. Mohamed, M.H.A. Awadalla, Hybrid algorithm for multiprocessor task scheduling, Int. J. Comput. Sci. Issues 8 (2) (2011).
[50] S. Dhingra, S.B. Gupta, R. Biswas, Genetic algorithm parameters optimization for bi-criteria multiprocessor task scheduling using design of experiments, Int. J. Comput. Control Quantum Inf. Eng. 8 (4) (2014).
[51] M.M. Fahmy, A fuzzy algorithm for scheduling non-periodic jobs on soft real-time single processor system, Ain Shams Eng. J. 1 (2010) 31–38.
[52] X. Kong, C. Lin, Y. Jiang, W. Yan, X. Chu, Efficient dynamic task scheduling in virtualized data centers with fuzzy prediction, J. Netw. Comput. Appl. 34 (4) (2010).
[53] K. Abda, K. Abharya, R. Mariana, Efficient scheduling rule for robotic flexible assembly cells based on fuzzy approach, Proc. Comput. Sci. 3 (2012) 483–488.
[54] G.A. Suer, A. Sarkar, A. Arinsoy, P.A. Kubi, K. Plis, M. Altun, Evaluation of feedback among multiple scheduler profiles in fuzzy genetic scheduling, Procedia Comput. Sci. 12 (2012) 129–134.
[55] A. Mehranzadeh, S.M. Hashemi, A novel-scheduling algorithm for cloud computing based on fuzzy logic, Int. J. Appl. Inf. Syst. 5 (7) (2013).
[56] Priya, N.K. Babub, Moving average fuzzy resource scheduling for virtualized cloud data services, J. Comput. Standards Interfaces 50 (2018) 251–257.
[57] X. Qin, H. Jiang, A dynamic and reliability-driven scheduling algorithm for parallel real-time jobs executing on heterogeneous clusters, J. Parallel Distrib. Comput. 65 (8) (2005) 885–900.
[58] Z.C. Papazachos, H.D. Karatza, The impact of task service time variability on gang scheduling performance in a two-cluster system, Simul. Modelling Practice Theory 17 (2009) 1276–1289.
[59] Q. Zhang, H. Liang, Y. Xing, A parallel task scheduling algorithm based on fuzzy clustering in cloud computing environment, Int. J. Mach. Learn. Comput. 4 (5) (2014).
[60] G.T. Abraham, A. James, N. Yaacob, Group-based parallel multi-scheduler for grid computing, Future Gener. Comput. Syst. 50 (2015) 140–153.
[61] V. Dandhwani, V. Vekariya, Multi-objective task scheduling using k-mean algorithm in cloud computing, Int. J. Innovative Res. Comput. Commun. Eng. 4 (11) (2016).
[62] D. Yokoyama, B. Schulze, H. Kloh, M. Bandini, V. Rebello, Affinity aware scheduling model of cluster nodes in private clouds, J. Netw. Comput. Appl. 95 (2017) 94–104.
[63] S. Abrishami, M. Naghibzadeh, Deadline-constrained workflow scheduling in software as a service Cloud, Sci. Iranica 19 (3) (2012) 680–689.
[64] S. Abrishami, M. Naghibzadeh, D.H.J. Epema, Deadline-constrained workflow scheduling algorithms for Infrastructure as a Service Clouds, Future Gener. Comput. Syst. 29 (2013) 158–169.
[65] S. Su, J. Li, Q. Huang, X. Huang, K. Shuang, J. Wang, Cost-efficient task scheduling for executing large programs in the cloud, Parallel Comput. 39 (2013) 177–188.
[66] J.H. Son, J.S. Kim, M.H. Kim, Extracting the workflow critical path from the extended well-formed workflow schema, J. Comput. System Sci. 70 (2005) 86–106.
[67] P. Senkul, I.H. Toroslu, An architecture for workflow scheduling under resource allocation constraints, Inf. Syst. 30 (5) (2005) 399–422.
[68] W. Wang, G. Zeng, D. Tang, J. Yao, Cloud-DLS: dynamic trusted scheduling for cloud computing, Expert Syst. Appl. 39 (2012) 2321–2329.
[69] L.D.D. Babu, P.V. Krishna, Honey bee behavior inspired load balancing of tasks in cloud computing environments, Appl. Soft Comput. 13 (5) (2013) 2292–2303.
[70] S. Lee, S. Kumara, N. Gautam, Efficient scheduling algorithm for component-based networks, Future Gener. Comput. Syst. 23 (4) (2007) 558–568.
[71] S. Ren, Multicriteria decision-making method under a single valued neutrosophic environment, Int. J. Intell. Inf. Tech. 13 (4) (2017) 23–37.
[72] S. Krishnamoorthy, G. Sudha Sadasivam, M. Rajalakshmi, K. Kowsalyaa, M. Dhivya, Privacy preserving fuzzy association rule mining in data clusters using particle swarm optimization, Int. J. Intell. Inf. Tech. 13 (2) (2017) 1–20.
[73] A.R. Arunarani, D. Manjula Perkinian, Intelligent techniques for providing effective security to cloud databases, Int. J. Intell. Inf. Tech. 14 (1) (2018) 1–16.
[74] T. Herawan, Y.M.M. Hassim, R. Ghazali, Functional link neural network with modified artificial bee colony for data classification, Int. J. Intell. Inf. Tech. 13 (3) (2017) 1–14.

AR. Arunarani is currently a Teaching Fellow at the Faculty of Computer Science and Engineering, CEG Campus, Anna University, Chennai, India. She received her Ph.D. from the Faculty of Information and Communication Engineering in 2018, M.E. degree from the Department of Computer Science and Engineering in 2011, and B.Tech. degree from the Department of Information Technology in 2008, all at the CEG campus, Anna University, Chennai. Her specializations include Networking, Data Mining, Big data and Cloud Computing. Her current research interests are Scheduling in Cloud, Network Security, and IoT.

Dr. D. Manjula is the Professor and Head of the Computer Science and Engineering Department, CEG Campus, Anna University, Chennai, India. She received her B.E. degree in Electronics and Communication Engineering from Thiagarajar College of Engineering, Madurai in 1983 and her M.E. degree in Computer Science and Engineering degree from Anna University, Chennai in 1987. She obtained Ph.D. degree in Information and Communication Engineering from Anna University in 2004. She has nearly 35 years of teaching experience in CEG campus, Anna University, Chennai, one of the top most State Government University in India since 1987. Her research interests spans over Cloud Computing, Social Network Analysis, Information Retrieval, Language Technologies, Data and Text Mining, Machine Learning, Imaging and Networks. Under her guidance, 11 Ph.D. Scholars are awarded and 9 Ph.D. scholars are doing their research in her areas of expertise. She has guided more than 50 post graduate students to complete their projects in her research fields. She has published over 130 technical national and international Journal as well as Conference papers. She has authored three books published by well-known publisher, Pearson Education Inc. She is the Convenor of CoMSE 2015 and DaSAA 2016. She is the life time member of ISTE. She has delivered many tutorials and Invited talks at different Universities and Institutions across India. She is currently coordinating Big Data Analytics Project under UGC-DSA Phase II. She has successfully completed the Database and Web Technology Project under the UGC-DSA Phase I. She has also coordinated three research projects funded by Tamil Virtual University of Tamil Nadu Government with a passion to render Tamil language in Mobile phones. She was actively involved in a project named RCILTS funded by Ministry of Information Technology, intended to provide knowledge tools in Indian languages (Tamil Language). Apart from the academic responsibilities, she was one of the Deputy Directors in the Centre for Distance Education. She has served as a Director, Planning & Development of Anna University which is directly related to the planning & development of every sector in the University. Moreover, she was the ex-officio member of all the executive committees constituted by the Vice-Chancellor. She has also acted as the Director, Centre for Faculty Development, Anna University. Through the Centre for faculty development, she has conducted numerous faculty development training programs.

Vijayan Sugumaran is Professor of Management Information Systems and Chair of the Department of Decision and Information Sciences at Oakland University, Rochester, Michigan, USA. He received his Ph.D. in Information Technology from George Mason University, Fairfax, Virginia, USA. He is also the Co-Director of the Center for Data Science and Big Data Analytics at Oakland University. His research interests are in the areas of Big Data Management and Analytics, Ontologies and Semantic Web, Intelligent Agent and Multi-Agent Systems. He has published over 200 peer-reviewed articles in Journals, Conferences, and Books. He has edited twelve books and serves on the Editorial Board of eight journals. He has published in top-tier journals such as *Information Systems Research*, *ACM Transactions on Database Systems*, *Communications of the ACM*, *IEEE Transactions on Big Data, IEEE Transactions on Engineering Management*, *IEEE Transactions on Education*, and *IEEE Software*. Dr. Sugumaran is the **editor-in-chief** of the *International Journal of Intelligent Information Technologies*. He is the Chair of the Intelligent Agent and Multi-Agent Systems mini-track for Americas Conference on Information Systems (AMCIS 1999–2019). Dr. Sugumaran has served as the program co-chair for the 14th Workshop on E-Business (WeB2015), the International Conference on Applications of Natural Language to Information Systems (NLDB 2008, NLDB 2013 and NLDB 2016) and 29th Australasian Conference on Information Systems (ACIS 2018). He also regularly serves as a program committee member for numerous national and international conferences.