International Journal of Engineering & Technology, 7 (3) (2018) 1677-1695

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET
doi: 10.14419/ijet.v7i3.13016
Research paper

Review of the quality of service scheduling mechanisms in cloud

K R RemeshBabu[1]*, Philip Samuel[2]

[1]Research Scholar, School of Engineering, Cochin University of Science and Technology, Kochi, India
[2]Professor, Department of Computer Science, Cochin University of Science and Technology, Kochi, India
**Corresponding author E-mail:remeshbabu@gecidukki.ac.in*

Abstract

Cloud computing provides on demand access to a large pool of heterogeneous computational and storage resources to users over the internet. Optimal scheduling mechanisms are needed for the efficient management of these heterogeneous resources. The optimal scheduler can improve the Quality of Services (QoS) as well as maintaining efficiency and fairness among these tasks. In large scale distributed systems, the performance of these scheduling algorithms is crucial for better efficiency. Now the cloud customers are charged based upon the amount of resources they are consumed or held in reserve. Comparing these scheduling algorithms from different perspectives is needed for further improvement. This paper provides a comparative study about different resource allocation, load balancing and virtual machine consolidation algorithms in cloud computing. These algorithms have been evaluated in terms of their ability to provide QoS for the tasks and Service Level Agreement (SLA) guarantee amongst the jobs served. This study identifies current and future research directions in this area for QoS enabled cloud scheduling.

Keywords:*Cloud Computing; Load Balancing; QoS; Resource Allocation; Task Scheduling; Service Level Agreement*.

1. Introduction

Cloud computing is a paradigm that enables on-demand network access to a shared pool of configurable virtual resources which can be rapidly provisioned and used based on the pay-per-use model. Cloud computing allows storing of data and accessing of computing resources such as processing power, data and programs over the internet instead of local computer's hardware. It is a form of distributed system based on virtualization technology.

Resource management in cloud computing infrastructure is handled by Virtual Machine (VM) scheduling and it will reduce operational as well as energy cost. The scheduling is the process of allocation of different tasks to resources with high quality, considering the parameters such as makespan, time skew, energy, cost, profit etc.

Now, cloud computing became the global computing infrastructure for business applications by providing large scale services with minimum cost. The ubiquitous nature with on demand computing facility made it as a popular computing model. It is a promising paradigm for the computing world that offers on-demand Information Technology resources and services to the customers over Internet. Since the users only need to pay for the services they actually used, there is a rapid growth in the usage of cloud resources. The main objective of this study is to review the different resource allocation, task scheduling and load balancing techniques in cloud computing based on various Quality of Service (QoS) parameters.

The cloud resources can be dynamically reconfigured to adjust variable load (scale), and it allows optimum resource utilization. These pools of resources are made available to the customers based on pay-per-use model which guarantees QoS as per customized Service Level Agreement (SLA).

In order to attract more customers, Cloud Service Providers (CSP) attempt to provide more sophisticated services with QoS. For ensuring QoS, CSPs need more accurate resource management services to process user submitted tasks. E.g. Amazon's Elastic Compute Cloud (EC2), provides an opportunity to auction based spot pricing. A client can use the services on the basis of bidding price.

There are several scheduling methods exist in the cloud computing, due to its multi-tenant, on-demand, elastic nature with pay-as-you-go model. The dynamicity of cloud in resource and task scheduling gives several opportunities to the researchers. Schedulers have to consider trade-off between functional as well as non-functional requirements in order to attract customer and QoS with profit.

1.1. Cloud service and delivery models

Cloud computing provides three distinct type of services namely Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

IaaS is typically, the process of allocating resources and virtual machines hosted in the cloud upon user request. These VMs are made available to the customers, and the customers have to maintain other required applications as they needed. Here the resource provisioning, task processing, data storage, network maintaining and management of other computing resources are done by the respective CSP.

PaaS includes web sites, web applications, etc. The user deploys the newly created or acquired applications using programming languages and tools supported by the provider.

SaaS offers applications like email, Customer Relation Management (CRM), cloud storage, etc. Here the applications are accessible from various client devices through an interface such as web

browser or a mobile device. The users can hire a software or application as they needed, instead of purchasing and installing in their personal computers.

There are different deployment models in cloud computing. A private cloud is used exclusively by one user. The service provider owns and operates the cloud infrastructure and service available in private access. A community cloud is used exclusively by a group of people or a community. It is owned by the members of the community or may be rented from service providers and the management is performed accordingly. A public cloud is used openly by the general public. The services in public clouds are available to the general public or a large industry group is owned by an organization selling cloud services. While a hybrid cloud is made up of two or more deployment models (private, community and public) within the same organization. It is providing the services of cloud and also on-premise offerings. Figure 1 represents the cloud computing model with different service and delivery models.

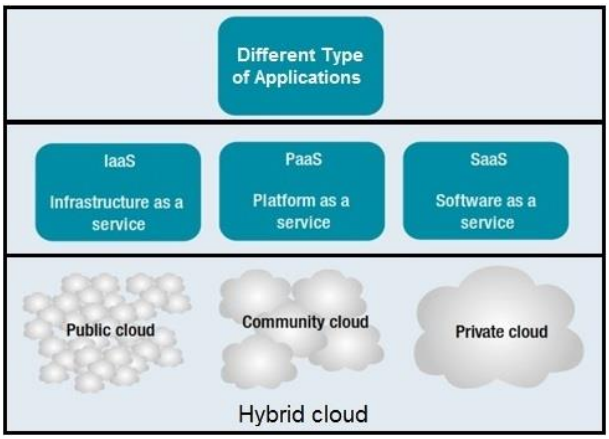

Fig. 1: Familiarization of Cloud Computing Model.

This paper presents a comprehensive survey and analysis of different resource allocation and scheduling methods proposed in cloud computing. It covers different types of resource provisioning, task scheduling, VM placement and load balancing methods based on the cloud properties. Other optimization methods like linear programming, meta-heuristic, heuristic, hybrid, elastic and multi-objective methods are also reviewed. The parameters considered, environment in which the algorithm is tested, highlights and limitations of each method is analyzed and compared in different tables. This overview will helps to get an in-depth knowledge about resource provisioning and scheduling schemes in cloud computing. Most of the resource management techniques is based on time and cost parameters and we observed that different QoS parameters affect the scheduling process. The parameter-based scheduling becoming more effective, especially when the cost is the primary factor. The heuristic and hybrid algorithms perform well with maximization of resource utilization and load balancing, since task scheduling is an NP-hard problem. In elastic cloud, migration and load balancing based algorithms have less completion time and energy consumption with better QoS.

The proposed schemes are classified into various groups based on type of algorithm, parameters considered, and number of objectives. This analysis helps to understand scope of the task scheduling and trends in the resource provisioning. In order to identify the issues of the cloud and to design and develop an effective task scheduling method, a comprehensive review is required about existing scheduling methods.

1.2. Taxonomy adopted

This review considered 118 papers out of which 60% of the papers are from last two years. The papers are selected based on the relevance in this field and citations received due to its excellence. The statistics of the papers selected for the review is given in Figure 2.

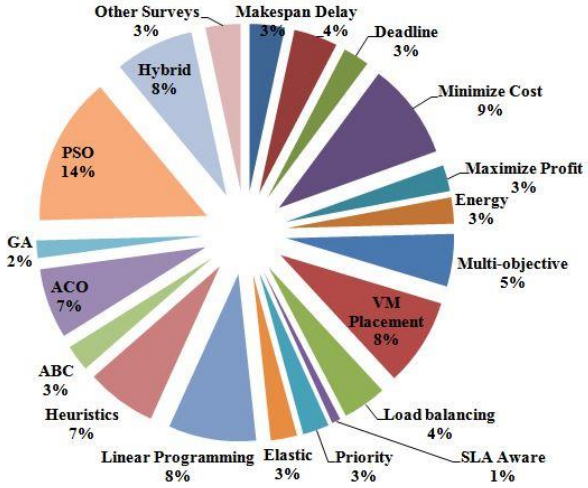

Fig. 2: Number of Papers Considered.

The rest of this paper is organized as follows: Section 2 explains about the significance of task scheduling and resource provisioning problem. Section 3 describes the cloud properties that affect scheduling. Section 4 covers the detailed comparative study of various scheduling models. The section 5 discuss about the optimization methods proposed in the cloud based on mathematical as well as natural algorithms. Observations and open issues in cloud computing are given in section 6. Finally section 7 provides the conclusion about this study.

2. Significance of scheduling

In cloud computing, resource management is an important task in scheduling of services, user tasks, and hardware infrastructure. The scheduling is the allocation of user submitted tasks to particular VM provisioned in a Physical Machine (PM).When demand increases from the user's side, then service provider can extend their computation resources beyond their boundaries in order to accommodate incoming requests. Cloud needs efficient intelligent task scheduling methods for resource allocation based on workload and time. Optimal resource allocation minimizes the operational cost as well as execution time. This in turn reduces power and energy consumption and operational cost. Hybrid technology is needed to support customers to choose different computation offers from CSPs. The offers from CSPs are attract customers to promote their business, and to reduce the operational cost. CSPs offers services in different categories such as subscription of services with expertise, SLA based, compliance, scalable and cost effective manner.

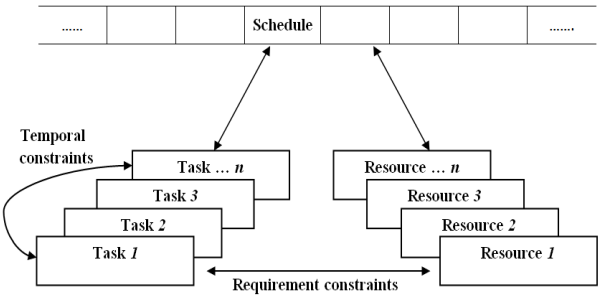

Fig. 3: Scheduling in Cloud.

The resource provisioning techniques decides which resources are to be made available to meet the customer requirements, while task scheduling is the process of allocating user tasks to the resources based on some criteria. Resource allocation is performed by scheduling of resources based on temporal and user requirement constraints. In the dynamic cloud environment, both user requirements and cloud resource status vary with time, hence scheduling based on temporal constraints is a cumbersome task.

So constraints plays major role in scheduling. Proper consideration of constraints will produce high level of QoS. Figure 3 gives an illustration of resource management with scheduling of services based on constraints in cloud.
To avoid unnecessary conditions, researchers have proposed several resource allocation strategies. A good resource allocation policy must avoid certain situations as follows.
Resource contention: it occurs when more than one user request for same service at the same time.
Scarcity of resources: it occurs when the availability of resource is limited.
Resource fragmentation: if the service provider can have enough resource to accept new request, but it is unable to allocate that request.
Over-provisioning: The application gets surplus resources than the demanded one.
Under-provisioning: The application is assigned with less number of resources than demanded.

3. Cloud properties that affect scheduling

3.1. Homogeneity

In a homogeneous cloud, the entire software stack including hypervisor, intermediate cloud stack and user portal are from the same service provider. So here management is simple, since the entire things are from a single provider. Since everything comes in a pre-integrated manner, if anything goes wrong, just one party holds the responsibility. When one CSP is in the possession of so much power, users become dependent on the same provider's technical and commercial strategy. The advantage of this kind of cloud environment is that, users can able to specialize in a CSP's tool. While administrators can easily cover for each other within this strategy, the downsides are different. The features are available on the technical side, but which is exclusively developed by the particular service provider. Besides, when a user is "locked in" to one service vendor strategy, resources can be easily delegated despite changes in the pricing structure. This belongs to the commercial side advantage.

3.2. Heterogeneity

In order to increase the performance and attract more customers, CSPs are adding different types of computing resources with increased memory and storage capacities. Thus heterogeneity improves the overall cloud performance and its power efficiency. Users are often looking for sophisticated high-end infrastructure such as high speed processors, with low cost. The moves towards green computing standards are now focusing on energy consumption. So public CSPs are now implementing different mixture of architectures for their infrastructure to improve power efficiency. This complex heterogeneous cloud data centre needs more powerful dynamic algorithms for resource and task management. Internets of Things (IoT) implementations are now rapidly increasing around the world. These IoT devices generate massive amount of data and need more processing power to analyse it. Hence heterogeneous cloud implementations are necessary for the successful IoT and related Cyber Physical Systems (CPS) implementations.

3.3. Elasticity

In cloud computing, elasticity is defined as the degree to which a system is able to adapt workload changes by provisioning and deprovisioning resources in an automatic manner such that, at each point in time the available resources match the current demand as closely as possible. Elastic cloud infrastructure provides a cloud computing environment with greater flexibility and scalability. Amazon Web Service (AWS) facilitate web service scalability.
Elasticity is the ability to fit the resources needed to cope with workloads dynamically usually in relation to scale out. When the load increases, adding more resources by scaling and when demand wanes, the system shrinks backs and removes unused resources. Elasticity is mostly important in cloud environments where pay-per-use and don't want to pay for resources that user do not currently need on the one hand, and want to meet rising demand when needed on the other hand. Elasticity adapts to both the "workload increase" as well as "workload decrease" by "provisioning and de-provisioning" resources in an "autonomic" manner. Intelligent algorithms that detect workload necessities will aid in this situation.

3.4. Scalability and auto scaling

Scalability is the ability of the cloud eco system to accommodate larger workloads by adding more resources either making hardware stronger (scale-up) or adding additional nodes (scale-out). Scalability is performed before the increase in workload by adding additional resources or to perform well before to meet the required QoS. This enables a CSP to meet expected quality demands from the customers or to meet SLA requirements for services with long-term, strategic needs. Auto scaling mitigates the resource contention and delay in processing user tasks. It aids CSPs to offer high level of services on demand with customer satisfaction. By scaling-out instances seamlessly and automatically when demand increases, better resource management can be done. By turning off unnecessary cloud instances automatically, CSPs can save money when demand reduces thereby achieves energy consumption. Also it can replace unhealthy or unreachable instances to maintain higher availability for user applications.

4. Scheduling models in cloud

The aim of a scheduling model is the allocation of resource in an optimal way to achieve better quality of service. Several classification of scheduling methods are available like static and dynamic, but this paper focus on the methods based on how resource provisioning, task scheduling, VM placement and load balancing are done. The general classification of our approach is given in Figure 4.

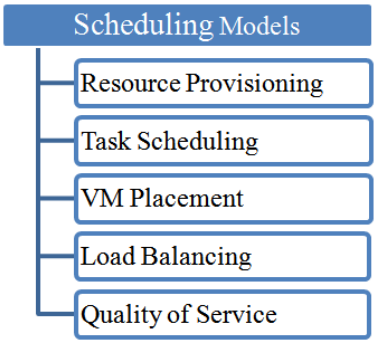

Fig. 4: General Classification of Scheduling Models.

4.1 Resource provisioning

Resource provisioning can be done in both static and dynamic environment. In static environment, the CSPs take resource allocation decision before the execution of user tasks. So if a workload requires more processing power, there doesn't exists any facility to add or acquire resources on the fly and it is forced to perform within the already allocated limited resource. This reduces the overall performance and the efficiency of cloud.
Dynamic method improves the resource allocation process by changing the initially allocated resources according to the needs of the workload. It is able to find resource requirement on the fly, and supports elasticity. These algorithms can work in both static and dynamic environment. Usually dynamism is fired based on some constraints like SLA, deadline requirements etc. Some cloud offers minimum QoS to the customers in order to cope with of-

fered QoS. The resource provisioning classification is shown in Figure 5.

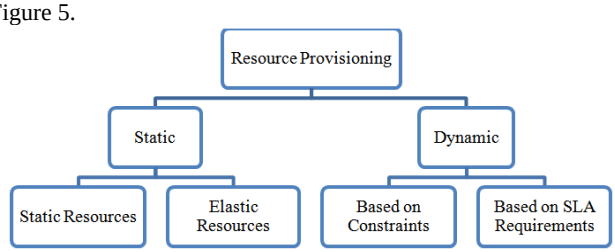

Fig. 5: Resource Provisioning Strategies.

4.2. Task scheduling

Task scheduling algorithms mainly focused on achieving some user specified objectives. These objectives are mostly user requested QoS parameters like makespan, deadline, response time, delay, cost, VM bandwidth, etc. Some of them focused on service level agreements between customer and provider. The commonly used CSP driven objectives are power and energy consumption with profit. The overall classifications of objectives are represented in the Figure 6.

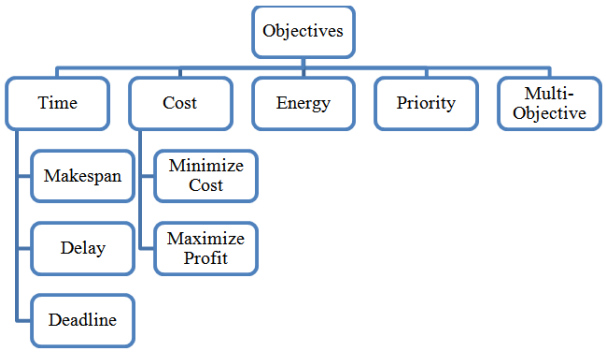

Fig. 6: Scheduling Strategies.

Makespan: It is the total completion time taken to complete a user submitted task. Most of the algorithms mentioned in this survey are focused on makespan as the important parameter.
Delay: It is one of the important factors in measuring quality of service. So delay in giving response to the customers is one of the parameter considered in this review.
Deadline: Usually the scientific workflows submitted to the cloud are to be completed within a specific time. This survey considered sufficient number of deadline constrained papers for the comparison.
Cost: The main objective of cloud is to minimize the cost of computation. The algorithms try to minimize the usage cost or try to provide more efficient service to the customers with amount they spend to hire the service.
Profit: While offering low cost services to the customers, CSPs are trying to maximize their revenue by attracting more customers. This is usually done by giving different offerings to the customers and maximizes their resource utilization rate.
Energy: Consumption of energy is crucial in reducing operational cost. One of the main cost incurring in running a cloud datacenter is energy cost. Most of the recently proposed methods are given keen attention to the power utilization and energy consumption.
Multi-Objective: The recent advancements in cloud scheduling methods have given attention to multiple criterions in task scheduling. These criterions are sometimes contradictory, so a trade off is needed between different solutions produced by the scheduler.

4.2.1. Makespan

In cloud, makespan is the total elapsed time from the submission of a task to the CSP. Different methods use diverse strategies to reduce makespan. It includes, execution time, delay in communications, response time, migration time etc. The major goal of scheduling task is to reduce completion time [85] and the maximum utilization of available resources [84][83]. Frequent migration of VMs also affects completion time. In real conditions there will be more complex interactions between different migrating tasks. Some independent migrations can be performed in parallel; other migrations in the system may share the same bottleneck link in their paths. This simultaneously leads to increase in migration time. In a large data centre, hundreds of migration requests can take place in a few minutes, where the effect of migration order becomes more significant. Therefore, a proper migration plan is needed to minimize the total migration time and to reduce imbalance. The details about makespan oriented methods are summarized in Table 1.

4.2.2. Delay

A task scheduler should consider the delay in processing of user submitted tasks and the depreciation while evaluating the CSP services. Queuing delay analysis proposed in [86] accounts for both delay-sensitive and delay tolerant applications. To obtain CSP's optimal pricing strategy, they formulated a profit maximization problem, which is non-convex in general. While in a multi-cloud allocations, additional delay and cost occurs due to inter-cloud communication [87] [88]. But its performance gains and cost savings are still significant than single cloud implementations. The economy of scale offered by cloud computing has attracted an increasing number of corporations to deploy their applications in cloud data centres. The uncertainty in the arrival of tasks brings a big challenge for a private cloud to schedule all the arrived tasks while guaranteeing the service delay bound [87]. The profit maximization problem can be solved by a Profit Maximization Algorithm (PMA) and it provides a temporal task scheduling, which can dynamically schedule all the arrived tasks that can be in private or public clouds [88]. Most of the existing scheduling algorithms are pre-emptive in nature. A number of context switching occurs in each pre-emptive scheduling [86] [88]. Context switching requires a certain amount of time and energy for saving and loading the registers and mapping the respective memory, updating various tables and lists, etc., which causes a hike in power consumption and delay during packet transfer. It also produces overhead in CPU and memory for storing and retrieving the details of a process at run time. The article [90] proposed a method that tried to mitigate these problems. Review findings on delay aware methods are given in Table 2. All the methods discussed in the computation offloading strategy is optimized to minimize the total energy consumption for both communication and computation while completing the submitted tasks within a given delay constraint. This is needed to reduce delay and improve completion time as well as operational cost.

4.2.3. Deadline

For a deadline constrained application, meeting the application's time limit requirement is critical, but there is no incentive to finish the application earlier. Hence, if a system can guarantee an application's deadline requirement with the least number of resources it will increase credibility of the provider. Further minimization in application's makespan under the least number of resources, both clients and service providers cost benefit will be higher [22]. Furthermore, as in a cloud environment, virtual machine instances are charged only when they are running. Such feature enables users to further reduce cost by running virtual machines intelligently [23] [24]. In [22] a new scheduling technique is proposed to schedule parameter sweep workflow which is dynamically executed in multiple instances. This novel method estimates the execution progress together with a workflow instantiation control and a cloud resource adjustment mechanism. The technique, with the objective to minimize cost within a deadline is evaluated using three existing task mapping heuristics. Their experiment results show that the proposed technique is able to lower cloud usage cost when the time constraint is relaxed. Table 3 gives summary of findings on deadline constrained methods.

4.2.4. Minimize cost

To minimize the cost of data placement for time varying workload applications, developers must optimally exploit the price difference between storage and network services across multiple CSPs. The volume of data that one has to collect and process for effective monitoring of datacenters possess significant big data challenges in collection, analysis, and storage.

With the increase in demand for cloud platform, the workload becomes more diverse and the one-fit-all pricing policy does not provide flexibility to the user. Sometimes it may provide poor user flexibility and energy efficiency. To keep up with the rapid evolution in information infrastructure a more flexible way of controlling cloud systems is proposed to satisfy the user and energy cost in [96].

The bidding strategies [100] based on three performance metrics such as cost, waiting time, and interruption rate is another cost based method. The provider allocates VM instances to the customers based on all the received bids, as well as on the computing capacity available. Users bidding above or below this published price are declared either successful or unsuccessful. Spot pricing creates an auction-based market for available cloud computing resources. Users can submit bids to the market at any time, using the spot price history to decide how much to bid. The cloud provider sets the spot price at regular time intervals, e.g., every five minutes, depending on the number of bids received from users (demand) and how many resources are available (supply) at each time slot [97] [101]. In these mechanisms, users' bids above the spot price are accepted, and that below is rejected in each time slot. Running spot instances [106] are terminated if their original bid prices fall below the new spot price, and relaunched only when their bids again exceed the spot price.

The explosive prevalence of IoT, big data, and fog computing makes the involved services and resource management makes more complicated than ever before. Due to resource limitations [101], resource heterogeneity [102], locality restrictions, environmental necessities and dynamic nature of resource demand, the resource allocation and scheduling is one of the essential problems, to be taken into account to adapt to the changing infrastructure environments [104]. The goal of resource allocation and scheduling is to maximize the efficiency of resources utilization, satisfy the users quality of service QoS requirements, meanwhile maximize the profit of both providers and users and so on, which presents a challenging problem. However, in respect of fog resource allocation [103], the current literature says at a shallow level of overview and exploration, almost no substantive research on the technology.

In the Petri net [103] based model, user's credit evaluation of CSP is taken as the primary parameter for task allocation. This novel market scheduling model considers cost of computing and take income discriminate function value as a decision making factor for task pre-emption. This market scheduler first schedules service-suppliers' tasks with worse credibility among users while realizing the income maximization of service suppliers so as to eradicate their bad impression of "income-oriented". Market oriented cloud is a new model of market balancing system for the participated consumer and provider in the online transaction. The need of market oriented cloud is to provide high QoS to the consumer and manage this quality during its life time [106]. A provider has to consider the different service quality parameter of individual consumers and that's way they can achieve the customer satisfaction. Market oriented cloud resource management is necessary to manage supply demand ratio of cloud resource to reach market equilibrium [98] [105] [106]. Cost minimization methods are summarized in Table 4.

4.2.5. Maximize profit

User demand for services or resources are increasing day by day; it is difficult to allocate resources to the user on demand in the absence of an optimal resource allocation method. In the on-demand cloud computing, if the resource requested by the user is unavailable, it will reduce provider's business as well as reputation. Cloud users can select the provider with satisfied needs and budgetary constraints. A challenge faced by cloud service provider is the designing of resource allocation techniques that will tackle the problem of Virtual Network Monitoring (VNM). Clients send numerous requests to reserve computational and network resources and expect their QoS conditions to be maintained through the request life time [107]. One of the main features that define a VNM policy is the window size selection scheme. The low-to-high technique methods [107][109] achieved the best performance in terms of the ratio of served connections while the high-to-low method had advantage in terms of resource utilization. The merits and demerits of profit oriented methods are given in Table 5.

4.2.6. Energy

The minimum frequency is a reasonable, feasible and more general model for expressing QoS requirements. Firstly, a task has a fixed number of instructions when it is allocated to a processor. The number of instructions that can be processed per unit time is different, which results in different execution time [92] [93] at different frequencies. Thus, this is why the frequency can represent the relative task execution time. In large environment, it is very difficult to measure the execution time with minimum frequency [91]. The bottleneck of some tasks is not the computing, but users may need a lower frequency to match the bottleneck of other resources, and avoid wasting the economic cost of computing resources. Service providers can dynamically or statically recommend prices for different frequencies so that the system can use Dynamic Voltage and Frequency Scaling (DVFS) based energy saving strategies [91] [92] to reduce electricity cost. Users could require tasks' running speed based on the price of services and the property of tasks. Service providers and users reached an agreement on energy-aware scheduling services. Moreover, the scheduler based on the improved ant colony algorithm supports multiple types of resource scheduling which meets the requirements of resource intensive applications in a real world scenario. The summary of the above methods are given in Table 6.

4.2.7. Multi objective

Multi-objective task scheduling algorithms have received particular attention in several researches and these methods are predominantly well suited to deal with multi-objective optimization problems. Also, some multi-objective meta-heuristics such as simulated annealing [16], Evolutionary Algorithms (EA) [17], Tabu Search (TS) and Particle Swarm Optimization (PSO) [13] have been proposed recently to address scheduling problems. There are also several efforts that move in the same direction that try to address smaller and simpler versions of scheduling. Due to high time complexity, Genetic Algorithm (GA) is not practical for large-scale applications. The algorithm introduced economic cost as a part of the objective function for data and computation scheduling, but does not consider storage constraints and cannot globally address the performance and cost optimization problem. A PSO-based heuristics [14] is another method to schedule applications to cloud resources that take into accounts both computation and energy cost. The main disadvantage of evolutionary algorithms is their high computational cost due to their slow convergence rate. In general, the cloud workflow scheduling is a complex optimization problem which requires considering different criteria so as to meet a large number of QoS requirements. The main contribution this approach for multi-objective workflow scheduling in clouds is to optimize the scheduling performance that incorporates energy consideration. Another method is based on the DVFS technique [13] to minimize energy consumption. This technique allows processors to operate in different voltage supply levels by sacrificing clock frequencies.

The main aim of a task scheduling strategy is to find a trade-off between user requirements and resource utilization. However, tasks which are submitted by different users may have different

requirements on computing time, memory space, data traffic, response time, etc. In addition, the resources which are included in cloud computing may be heterogeneous and geographically distributed. One of the problems in meta-heuristic method is the ability to avoid getting stuck with sub optimal solutions. In GA, mutation process is used to solve the problem. Ant colony algorithms allow the pheromone to evaporate in order to force the search of a new path. On the other hand, bee colony algorithms uses scout bee agents to randomly generate a new solution when a solution can no longer be improved after some iterations [15]. The summarized information about multi objective methods is shown in Table 7.

Table 1: Makespan

Paper	Method	Parameters	Highlights	Limitations	Environment
[83]	Fully Polynomial Time Approximation Algorithm (FPTA)	Migration time Computation time Transmission rate Bandwidth	Load balancing Low transmission rate	High SLA violations	Simulation
[84]	VM migration algorithm	Cost Migration time Resource utilization time	Load balancing Maximize resource utilization Minimum service interruption	Inefficient	Simulation
[85]	Cloud based Workflow Scheduling (CWSA)	Completion time Cost	Minimum completion time (MCT)	Service interruptions	Simulation
[1]	Map reduce framework scheduling in Hadoop	Completion time/Makespan Workload	Dynamic slot configuration feedback Control-based workload estimation	Sub optimal solutions No load balancing	Real

Table 2: Delay

Paper	Method	Parameters	Highlights	Limitations	Environment
[86]	Pricing algorithm	Profit Delay Cost	Delay tolerance	High energy consumption No SLA	Simulation
[87]	Resource allocation algorithm	Delay Cost	SLA constraints Multi-cloud resource allocation	No priorities No load balancing	Simulation
[88]	Profit maximization, SA-PSO	Delay Cost Profit	Profit maximization Delay bound	Service interruption	Simulation
[89]	VM scheduling algorithm	Delay Buffer size Power	Minimum delay Minimum power consumption High QoS	No load balancing Homogeneous resources No load balancing	Real
[90]	Computation offloading with energy constraints	Delay Communication cost Computation cost Energy	Delay tolerance Minimum energy consumption	Frequent service interruption Unreliable	Simulation

Table 3: Deadline

Paper	Method	Parameter	Highlights	Limitations	Environment
[22]	Minimal Slack Time and Minimum Distance (MSMD) algorithm	Execution time Cost	Minimize makespan Instance hour minimization Auto-scaling	Low efficiency	Simulation
[23]	Min-Min algorithm Heuristic algorithm	Execution time Cost Deadline	Optimized parameter-based sweep workflow	High execution time	Simulation
[24]	Heuristic algorithm Minimum Average Cost First (MACF)	Time Cost	Time slot filtering Greedy and fair-based scheduling	Pricing interval not considered No load balancing	Simulation

Table 4: Minimize Cost

Paper	Method	Parameters	Highlights	Limitations	Environment
[96]	Optimal offline & deterministic online algorithm	Cost Workload	Reducing horizon control	No load balancing	Simulation
[97]	Map reduce Offline simple task scheduling Online co-scheduling	Cost Makespan	Cost optimality Cost efficiently co-scheduling Flexibility in fine tuning the cost performance tradeoffs	Slow performance	Simulation
[98]	Dynamic Data Allocation Advance (2DA) algorithm Cost-aware heterogeneous cloud memory model	Cost Time	Reduction in operational cost	No load balancing	Simulation
[99]	Spot and dynamic pricing	Cost Resource use Waiting time Interruption rate	High biding option in online market	Performance overhead	Simulation
[100]	Bidding strategy Map reduce	Spot price Bid price	Optimal bidding	Interruption overhead	Simulation
[101]	Multi-criteria decision making	Job running time Bid price	Optimal cost saving	No automated tool for	Simulation

	framework Petri net based framework	Spot price Cost/benefit ratio	Reduce execution time	feedback No load balancing	
[102]	Bayes classifier design dynamic task scheduling algorithm	Cost Execution time Waiting time Deadline	Minimize execution time Minimize operational cost	Interruption rate is high	Simulation
[103]	Priced Timed Petri Net (PTPN) resource allocation strategy	Completion time Cost	Pre-allocated resources Credibility evaluation High cost saving	No load balancing	Simulation
[104]	Pre-emptive scheduling Schedule Model in a Cloud Computing based on Credit and Cost (SMCC)	Cost Task penalty Credit price Credibility	Discriminating function Maximization of service supplier	Interruption overhead No fairness among tasks	Simulation
[105]	Paddy Field Algorithm (PFA) Price detection algorithm	Cost Execution Time	Combinatorial double auction policy Better service satisfaction	Need balancing of bid price and spot price	Simulation
[106]	Holistic brokerage model	Cost	Scalability SLA negotiation	No Quality of Experience (QoE) Underutilization of resources	Simulation

Table 5: Maximize Profit

Paper	Method	Parameters	Highlights	Limitations	Environment
[107]	Mixed Integer Non Linear Programming (MNLP) formulation	Profit Service Penalty	Server consolidation Heuristic method	High SLA violations Slow No load balancing	Simulation
[108]	Price detection algorithm Cooperation Competition	Revenue Profit	Minimum energy consumption Revenue maximization	Network latency Delay No load balancing	Real
[109]	Profit driven optimization	Profit Execution Time	Scalability	Delay in service Low makespan	Simulation

Table 6: Energy

Paper	Method	Parameters	Highlights	Limitations	Environment
[91]	DVFS Bin packing algorithm	Execution time Cost Energy Frequency	Minimum energy consumption ratio (ECR) Minimum worst-case execution time (WCET)	Low efficiency	Simulation
[92]	DVS Energy-aware Dynamic Task Scheduling (EDTS)	Energy Execution time Cost	Minimum energy consumption Reduce cost	Lack of QoS support No load balancing	Simulation
[93]	PreAnt policy Bin packing algorithm	Energy Execution time	Manage instantaneous peak load Resource intensive application with QoS	Service interruption	Simulation
[94]	Optimal resource allocation with predetermined task placement & resource allocation algorithm	Energy Cost Job completion time	Increase utility and productivity Linear programming method	No load balancing Performance degradation	Simulation
[95]	Lagrange relaxation based Aggregated Cost Algorithm (LRAC)	Energy Delay Deadline	Collaborative task execution One-climb policy Minimum energy consumption	Low efficiency No load balancing	Simulation

Table 7: Multi-objective

Paper	Method	Parameter	Highlights	Limitations	Environment
[13]	PSO, DVFS & HEFT algorithm	Cost Time Energy	Workflow Scheduling Energy consumption	Low efficiency Sub optimal response time	Simulation
[14]	Nested PSO-based multi-objective task scheduling algorithm	Energy time	Energy optimization	Lack of service availability Frequent migrations	Simulation
[15]	ABC Algorithm	Cost Execution time Energy	Optimization in time and cost	Lack of workload management Frequent migrations	Simulation
[16]	Multi-objective cat swarm optimization with SA	Time Cost	Scalability	Low efficiency Slower Sub optimal solutions	Simulation
[17]	Multi-objective Evolutionary Algorithm (MEA)	Waiting time Cost Energy	Minimize energy consumption Optimization of cost and time	Low efficiency Slower Sub optimal solutions	Simulation
[18]	Min-Min based time and cost trade off algorithm	Time Cost	Multi-objective optimization model	Lack of failure recovery	Simulation

4.3. VM placement

VMs are the major interface to the resources where users run their applications on IaaS clouds. Many cloud providers allow users to create/maintain their own VM images (VMI), and even

buy/share/sell their VMIs (e.g., on Amazon EC2). Basically, users can control software installations and maintenance inside the VMI. The design and implementation of VMI management systems is challenging, due to the scale, complexity, variety and dynamics of VMIs. First, the prosperity of cloud computing creates rapidly growing users and diversifying applications. This leads an ever-increasing number of VMIs that are created and shared on the IaaS cloud. The VMI content can be stored as a file, a block device, a logical volume, a root partition or a complete hard disk drive. The adaptive spread-based policy uses an adaptive threshold to differentiate the long requests from the short, while scheduling short requests at the earliest possible slot and spreading long requests over the slots as even as possible. To achieve energy savings and emissions reduction, server consolidation technology using virtualization is introduced [34]. This technology can consolidate multiple applications on the same physical machine, with each application typically running on its own virtual machines. In return, these virtual machines are mapped to physical machines. In the context of virtualized data centres, it is a critical concern to design energy efficient virtual machine placement approaches that reduce energy consumption while satisfying cloud services/applications [39][40][41].

As the cloud computing progress the customers are demanding low cost efficient computation [37]. So the cloud providers have to minimize their computation cost [36] including power, energy usage, etc. Resource procurement can be accomplished using conventional or economic models. The conventional models assume that resource providers are non-strategic, whereas economic models assume that resource providers are rational and intelligent. In conventional methods, a user pays for the consumed service. In economic models, a user pays are based on the value derived from the service. Hence cost aware VM placement models are more appropriate in the context of cloud. The details are summarized in Table 8.

4.4. Load balancing methods

In order to cope with SLA agreements and to maintain QoS guaranteed, CSP have to adopt suitable load balancing mechanism across their computational resources. Load balancing mechanisms are trying to avoid overloaded and under-loaded conditions in the physical machines in a data centre. Too much load will degrade the overall performance, while under loaded conditions will results in high power consumption, energy and cost. Cloud computing becomes a well-adopted computing paradigm with the unprecedented scalability and flexibility. The data centre cloud is a new cloud computing model that uses multi data centre architectures for large scale massive data processing or computing. In data centre cloud computing, the overall efficiency of the cloud depends largely on the workload scheduler, which allocates clients' tasks to different cloud data centres. Developing high performance workload scheduling techniques [28] in cloud computing imposes a great challenge which has been extensively studied by several researchers. Most of the previous works aim only at minimizing the completion time of tasks. However, timeliness is not the only concern, while reliability and security are also very important. A comprehensive QoS model is proposed to measure the overall performance of data centre clouds. The load-balanced scheduling focuses on evenly distributing traffic among all links in a data centre network to enable the network to transmit more data flows with lower average end-to-end transmission delay. Traditional hardware based load balancing techniques cannot be widely used due to the high cost and the deficiency in programmable ability. Therefore, more and more researchers pay more attention on software-defined networking (SDN) techniques (e.g., OpenFlow) [29] that can improve transmission capacity of data centres through programmable load balanced flow control. The live VM migration is a technique for achieving system load balancing in a cloud environment by transferring an active VM from one physical host to another. This technique has been proposed to reduce the downtime for migrating overloaded VMs, but it is still time and cost consuming, and a large amount of memory is involved in the migration process. A Task Based System Load Balancing method using Particle Swarm Optimization (TBSLBPSO) [30] that achieves system load balancing by transferring only extra tasks from an overloaded VM instead of migrating the entire overloaded VM. There are several optimization models to migrate and balance workload across data centre to improve computation [31]. Load balancing mechanisms also have to limit frequent migrations in the system. Frequent migrations will create imbalance in the system, and affect performance adversely. Some of the mechanisms that consider imbalance in load balancing are Interference aware prediction mechanism [117], and enhanced bee colony [118] tries to reduce it. Summary of the findings about load balancing methods are tabulated in Table 9.

4.5. Quality of service

4.5.1. SLA aware

SLA aware Task Scheduling provides service with high quality service to the customers. In hybrid cloud infrastructure, the primary objective of a scheduler is to harmonize the SLA and to minimize the infrastructure as well as operational cost.

However, prior to acquiring a cloud service, cloud consumer needs to analyse the risk associated with adopting a cloud-based solution for particular information system, and plan for the risk-treatment and risk-control activities associated with the cloud based operations of a system. A hybrid cloud scheduling algorithm can be used in an elastic autonomous service network to solve these issues [2]. To do so, a cloud consumer needs to gain the perspective of the entire cloud ecosystem that will serve the operations of their cloud-based information system. For successful adoption of a cloud-based information system solution, the cloud consumer must be able to clearly understand the system's cloud-specific characteristics, the architectural components for each service type and deployment model, and the cloud actors' roles in establishing a secure cloud ecosystem. Understanding the relationships and interdependencies between the different cloud deployment models and service models is critical to identify the security risks involved in cloud computing. The differences in methods and responsibilities for securing different combinations of service and deployment models present a significant challenge for cloud consumers. They need to perform a thorough risk assessment to accurately identify the security and privacy controls necessary to preserve their environment's security level as part of the risk treatment process, and to monitor the operations and data after migrating to the cloud in response to their risk control needs. Table 10 gives the summary of SLA aware methods.

4.5.2. Priority

The priority is an important parameter to schedule the services in cloud [20]. The Memetic Algorithm (MA) is a class of optimization algorithms whose structure is characterized by an evolutionary framework and a list of local search components. The memetic algorithm in [19] merges together concepts from different search methodologies, and most prominently concepts from local search techniques and population-based search. To improve local search, first select an appropriate solution for a randomly specified local search direction and then apply local search only to the selected solutions. The advantage in optimization problems with memetic is that, it developed a static task scheduling on cloud environment using multiple priority queues. In a cloud computing environment, multiple customers are submitting job request with their constraints, i.e., multiple users are requesting same resource. For example, in a high performance computational environment which mainly deals with scientific simulations such as weather prediction, rainfall simulation, Monsoon prediction and cyclone simulation etc., requires huge amount of computing resources such as processors, servers, storage etc. Many users are requesting these computational resources to run their model which is specifically used for

scientific predictions.In this situation it will be a problem for cloud administrator to decide how to allocate the available resources among the requested users for minimize makespan and utilize resource effectively [21]. Summary table for above methods are provided in Table 11.

4.5.3. Elasticity-based

In economics, elasticity is the measurement of how responsive an economic variable is varying with another. In particular, elasticity can be quantified as the ratio of the percentage change in one variable to the percentage change in another variable. Using this definition, elasticity in cloud computing can be defined as how the amount of computing resource changes with the current workload. It seems that the definition is quantitative and measurable; however, such a definition of responsiveness is not entirely adequate, since it only considers how much, not how fast, the computing resource adapts. If a cloud computing platform takes a long time to provide the correct amount of resources to match the workload (which might not be current any more), it is not considered as elastic. Elasticity is meaningful to the cloud users only when the acquired VMs can be provisioned in time within the user expectation. The long unexpected VM start-up time could result in resource under-provisioning, which will inevitably hurt system performance [26]. Similarly, the long unexpected VM shut-down time could result in resource over-provisioning, which will inevitably hurt resource utilization. The auto-scaling capability of the cloud can ensure the service with QoS with minimizing the makespan and cost [27]. Table 12 gives the summary of elastic methods in cloud scheduling.

5. Optimization methods

Another classification of scheduling method is based on the optimization policies used in the algorithms. The dynamic nature of the cloud environment makes task scheduling as a cumbersome task. Scheduling in the dynamic cloud environment is NP-hard, so finding an optimal solution for the task assignment is difficult. Also the solutions are obtained by taking several assumptions on state of the cloud eco system. Nature inspired algorithms are capable to produce good sub optimal solutions using heuristics. Heuristics used by ants, bees, and flock of birds are some of the examples. The sub optimal category of algorithms can be further classified into heuristic, meta-heuristic and hybrid algorithms, based on how they are applied in the application scenario.

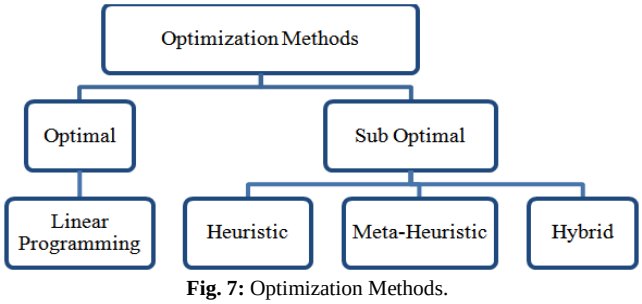

Fig. 7: Optimization Methods.

5.1. Linear programming model

Linear programming (also called linear optimization) is a method to achieve the best outcome (such as maximum profit or lowest cost) in a mathematical model whose requirements are represented by linear relationships. It is a special case of mathematical programming (mathematical optimization). In cloud, task scheduling is based on the linear method to obtain optimal solution. An agent is a computer system that is capable of making decisions independently, carrying out actions autonomously, and interacting with other agents through cooperation, coordination (achieving the state in which their actions fit in well with others), and negotiation (trying to reach agreements on some matters) [3]. Bin packing and heuristic algorithms are used for the analysis of real time visual data, image and video; but it requires lack of QoS. For making a comprehensive and efficient decision that assigns optimal resource capacity to the created VM, complicated correlations among reliability, performance and energy must be taken into account.

Cloud computing provides immense computing power with reduced cost. User can outsource their vast computational work to the cloud and use massive computational power, storage, software, network etc. Despite of all these benefits, there are still few obstacles in cloud computing regarding confidentiality and integrity of data [10]. Outsourcing and computation compromises the security of data being stored on cloud. Considering cloud as insecure platform an intelligent machine learning based VM allocation, that provide scalability as well as QoS is designed in[11]. To protect the data outsourced on cloud, encryption of sensitive data is the primary procedure. Encryption helps to maintain confidentiality of data stored in the cloud [12]. Other than encryption, the cloud itself is sometimes not very faithful which may lead to incorrect results. It is possible that software bugs, hardware failure or even outsider attack may decrease the result quality. A fault tolerant system with less power consumption is created by a Bayesian approach can generate the optimal result [5]. Summary of the above methods are shown in Table 13.

5.2. Heuristic methods

The term heuristic is used for algorithms which find solutions among all possible ones, but they do not guarantee that the optimal result will be found; therefore they may be considered as approximate algorithms. These algorithms, usually find a solution close to the best one and they find it fast and easily. The method used from a heuristic algorithm is one of the known methods, such as greediness, but in order to be easy and fast the algorithm ignores or even suppresses some of the problem's demands. It is designed to solve problems in a faster and more efficient manner than traditional methods by sacrificing optimality, accuracy, precision, or completeness for speed. Heuristic algorithms are usually used to solve NP-complete problems. Heuristics can generate a solution individually or be used to provide a good baseline and are supplemented with optimization algorithms but in certain situation it has no optimal efficiency [44]. PSO is one of the best heuristic algorithms that can be combined with resource selection algorithm to generate optimal energy consumption solutions [47].

Heterogeneous computing environments provide scalable computing resources for various applications, which are constructed by interconnecting machines with distinct processing capacity via different networks. Workflow scheduling in heterogeneous computing environments aims at assigning tasks to machines and achieves a highly efficient computing.

Modern problems tend to be very intricate and relate to the analysis of large data sets. Hybrid algorithms can be used as the combination of more than one heuristic algorithms to solve the existing problem [44][50][51]. Even if an exact algorithm can be developed, its time or space complexity may turn out unacceptable. But in reality, it is often sufficient to find an approximate or partial solution. Such admission extends the set of techniques to cope with the problem. There are few articles that discuss heuristic algorithms which suggest some approximations to the solution of optimization problems. In such problems the objective is to find the optimal of all possible solutions by minimizing or maximizing the objective function [43] [45] [48]. The objective function is a function used to evaluate a quality of the generated solution. Many real-world issues are easily stated as optimization problems. The collection of all possible solutions for a given problem can be regarded as a search space, and optimization algorithms are often referred to as search algorithm. Heuristic methods are covered in Table 14.

Table 8: VM Placement

Paper	Method	Parameter	Highlights	Limitations	Environment
[32]	Common Deployment Model (CDM)	Time Bandwidth Memory	Maximize resource utilization Use of active and passive directory	Unable to handle network latency	Simulation
[33]	Adaptive spread based scheduling algorithm	Bandwidth Cost Response time	Slicing scheduled tenant request model Maximize acceptance rate Minimize power usage rate	Low efficiency Slow Low response time	Simulation
[34]	Discrete PSO	Response time Cost	Maximize resource utilization Minimize energy consumption	Less reliable Low response time	Simulation
[35]	MigrateFS algorithm	Cost Execution time	Optimization model Scalability Detecting SLA violation	Low performance	Simulation
[36]	VM resource dynamic scheduling algorithm	Price Bandwidth	Resource utilization Minimize pricing	Low performance No load balancing	Simulation
[37]	Bin packing algorithm	Cost Profit	Best-fit and Worst-fit method Reduced SLA violations	No load balancing	Simulation
[38]	Greedy & PSO Algorithm	Completion Time Cost	Convergence rate is optimized Reduced completion time	No load balancing	Simulation
[39]	PSO	Energy	Energy efficient VM placement	No load balancing No SLA	Simulation
[40]	Improved PSO	Time	Increased resource availability	No load balancing No SLA	Simulation
[41]	Hybrid discrete PSO	Cost Energy	Energy efficient VM placement	Frequent migrations	Simulation

Table 9: Load Balancing Methods

Paper	Method	Parameter	Highlights	Limitations	Environment
[28]	Advanced Cross-Entropy based Stochastic Scheduling	Service rate Arrival rate	Scalability Flexibility Optimize QoS	Delay Frequent migrations	Simulation
[29]	Static offline optimal algorithm Network Overhead Minimization Algorithm	Bandwidth	Minimize inter-datacenter network load reduction	Low efficiency Delay	Simulation
[30]	Task-Based System Load Balancing (TBSLB)	Execution time Transfer time Cost	Pre-copy process maximizes resource consumption	Delay Frequent migrations	Simulation
[31]	Two stage load balancing	Cost Power	Pareto optimality	Low performance Delay	Simulation
[118]	Enhanced Bee colony algorithm	Makespan Cost	Load balancing	Delay in scheduling	Simulation

Table 10: SLA Aware

Paper	Method	Parameter	Highlights	Limitations	Environment
[2]	Hybrid cloud scheduler algorithm	Cost Deadline	Elastic autonomous service network	No load balancing	Simulation

Table 11: Priority

Paper	Method	Parameter	Highlights	Limitations	Environment
[19]	Memetic - GA method	Makespan Speed	Optimization Earliest finishing time	Delay No load balancing	Simulation
[20]	Priority algorithm	Time Cost	Maximum profit Minimum wastage of resources	Frequent migrations Low response time	Simulation
[21]	Min-Min algorithm Priority-based scheduling	Makespan Cost	Scalability Load balancing	Less fault tolerance Frequent migrations	Simulation

Table 12: Elasticity-based

Paper	Method	Parameter	Highlights	Limitations	Environment
[25]	Open Cloud Computing Interface (OCCI)	TimeCost	Autonomic loop	Multiple autonomic loop	Real
[26]	On-site elastic algorithm	Execution timeCost	Multi-level QoS service	Performance degradation & Delay Frequent migrations	Simulation
[27]	Dynamic Fault-Tolerant Scheduling (FASTER) Algorithm	Execution Deadline	Primary backup-based scheduling Auto scaling Backward shifting Resource utilization	Delay No load balancing	Simulation

5.3. Meta-heuristic methods

Heuristic algorithms are good for specific applications and it gives optimal solutions within a specific time. Meta-heuristic algorithms are computationally complex than heuristic algorithms, and more

suited for general purpose problems. But in elastic computing, where resources are unbounded and environment is challenging, meta-heuristics are good solution for obtaining optimal solutions.

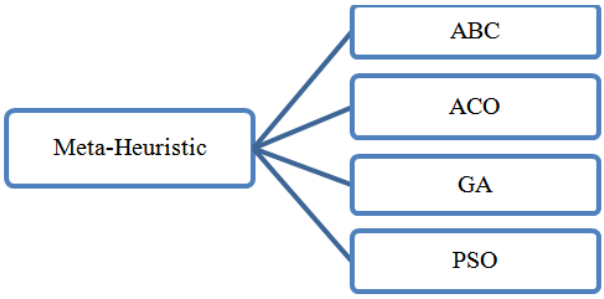

Fig. 8: Meta-Heuristic Methods.

5.3.1. ABC-based task scheduling

Artificial Bee Colony (ABC) algorithm was suggested by Karaboga. This method is inspired by the foraging behaviour of honey bees. It uses three kinds of honey bee in order to search food source: scouts bees, employed and onlookers. The position of a food source in ABC model represents a solution to the optimization problem and the number of employed or onlooker bees represents the number of solutions in the studied population. The fitness of the associated solution refers to the nectar amount of food source. At the beginning, the scout bees initialize the population randomly and after all employed bees finish the search of food sources, they share the information about nectar direction of the food source and distance from the hive with the onlooker bees. These later pick the best food source by evaluating the nectar quality and the distance to cross. Thus, the scout bees return back to the food source position to bring nectar to the hive.
The algorithm gives efficient performance as it uses both global exploration search and local exploitation search [53] [54]. Swarm intelligent and nature-inspired algorithms [55] are built for multi-objective optimization problems and are widely used in routing applications, data clustering, engineering design problems, medical image processing or specialized job scheduling algorithms [56]. These findings are summarized in Table 15.

5.3.2. ACO-based task scheduling

Ant Colony Optimization (ACO) is based on real ant's life. The algorithm was invented by Dorigo in 1992, this is called as ant system. An ant can find the shortest path between their food and colonies. It produces a pheromone and leaves it into the way they travel. The intensity of pheromone increases when more ants travel on same way. Then find out the shortest path based on the intensity of the pheromone. The ACO method [57] is helpful for solving Knapsack problem, travelling salesman problem, task scheduling problems in grid computing and cloud computing etc. The main concept of ACO is to simulate the searching behaviour of artificial ant's colonies. When group of ants searching for food, they secrete special kind of chemical known as pheromone. Firstly, ants randomly start to search their food. When food source is found, they spilt pheromone on the path; ants track the trails of the previous ant's food source by sensing pheromone on the soil. As this procedure continues, majority of the ants pull towards to choose the best-so-far path [60] [61] as there have been huge amount of pheromones accumulated on this path. Ants construct solutions to scheduling problem [57] [58] [59] during an iteration by moving from one VM to another until the tour is completed. ACO based methods are tabulated in Table 16.

5.3.3. GA-based task scheduling

Genetic Algorithm (GA) is a search algorithm which is based on the principles of evolution and natural genetics. It combines the exploitation of past results with the exploration of new areas of the search space. By using survival of the fittest techniques combined with a structured yet randomized information exchange, a GA can mimic some of the innovative flair of human search [65]. A generation is a collection of artificial creatures (strings). In every new generation, a set of strings is created using information from the previous ones. Occasionally a new part is tried for good measure. GAs are randomized, but they are not simple random walks. They efficiently exploit historical information to speculate on new search points with expected improvement. Several algorithms have been introduced to solve timetabling problems. The earliest sets of algorithms are based on graph colouring heuristics. These algorithms show a great efficiency in small instances of timetabling problems, but are not efficient in large instances. Later, stochastic search methods, such as GAs, SA, TS, etc., were introduced to solve timetabling problems [66]. The tabular information about GA based methods are given in Table 17.

5.3.4. PSO-based task scheduling

The concept of particle swarm is originally designed to find solutions for continuous optimization problems without prior information. To solve the workflow scheduling problem using conventional PSO, the key issue is to define the position and velocity of particle as well as to define their operation rules and the equation of motion according to the features of discrete variables. There are multiple objectives that needed to be satisfied in cloud systems including (performance, profit, and utilization). To address the problems with multiple objectives, a number of researchers have developed techniques for multi-objective optimization (MOO) [69][73][111]. Specifically, MOO studies the search methods that are used to find solutions based on several conflicting objectives such as performance in terms of minimizing waiting time and maximizing resource utilization or maximizing profitability. The PSO-based scheduling policy balance the load across the data centre [39] [40] [41][111][116]. Some of them focus on computation time, deadline, energy and profit [110] [112][115]. Table 18 gives summary about PSO methods.

5.4. Hybrid methods

Hybrid methods are combination of heuristic algorithms that obtain the optimal solutions for NP-hard problems in a cost effective manner with minimum execution time. Hybrid algorithms can be classified into four categories as shown in Figure 9. Temporal Task Scheduling Algorithm (TTSA) is an example for optimizing the throughput by using hybrid methods [75] in cloud. The Table 19 summarizes the different hybrid methods in cloud resource and task management.

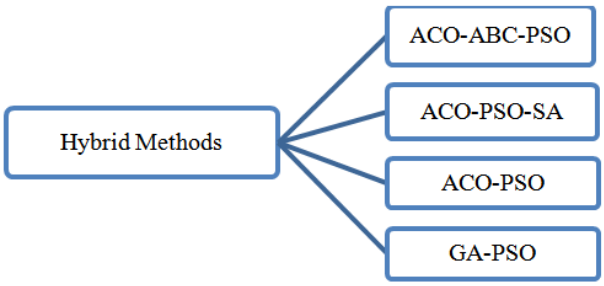

Fig. 9: Hybrid Methods.

5.4.1. ACO-ABC-PSO task scheduling

To improve the efficiency of scheduling process, currently available literatures are considering different parameters. These performance parameters are makespan, scalability, throughput, cost, resource utilization rate, fault tolerance, migration time and delay. A cloud task scheduling based ABC, PSO and ACO approaches [76] are proposed for the allocation of incoming jobs to VMs with considering the makespan parameter to achieve a high user satisfaction. The cloud task scheduling based on ABC PSO and ACO algorithm can achieve good system load balance than random and

FCFS. Sometimes increase in makespan leads to the increase in degree of imbalance and the reduction leads to the reduction in degree of imbalance. This is the reason of how ABC, PSO and ACO can achieve better load balancing than random and FCFS algorithms.

5.4.2. ACO-PSO-SA task scheduling

Multi-objective task scheduling problem should account consumers' QoS expectations. Cloud Scalable Multi-objective (CSM) task scheduling and optimization algorithm [77] considers execution time and cost. The novelty of the proposed method is that its design enhances the local search procedure of the algorithm in exploring larger search space that returns better optimum solutions.

5.4.3. ACO-PSO task scheduling

In order to get more desired results, hybridization of algorithms is one of the best solutions. Nature inspired algorithms can easily combine with classical algorithms or with other heuristic algorithms, which gives better results. The hybrid algorithms mentioned here are based on response time [78], artificial intelligence network load balancing using ACO [79] and with modified GA [80]. The combination of Throttled and Equal (TE) load share algorithm and Round Robin (RR) reduction scheme that will also help to gratify the request of customer services.

5.4.4. GA-PSO task scheduling

Workflow scheduling based on QoS constraints in cloud computing is an intractable problem. The hybrid PSO is suitable for continuous areas. The crossover strategy and mutation strategy of the GA is embedded into PSO, so that it can play a role in the discrete problem.The method [81] hybrid with particle swarm algorithm greatly improves the solution quality, so it can be used as an effective way to solve the cost minimization problem with due dates in cloud computing. It optimizes computation cost and data transmission cost. The PSO algorithm can also used for workflow scheduling applications.

In Cuckoo Search (CS) algorithm [82], an initial set of nests, which represent the solutions, are randomly generated. Then these solutions are updated over multiple generations. The new solution can replace a different randomly chosen solution if it has a fitness value better than the original. After this possible replacement of a solution, all of the nests are ranked by fitness and the worst fraction of the nests is replaced with random solutions. This combination of mechanisms allows the solutions to search locally and globally at the same time for the optimal solution.

Table 13: Linear Programming Model

Paper	Method	Parameter	Highlights	Limitations	Environment
[3]	Intelligent agent based approach	Price Availability Time	Agent-based computing Event condition action	No interoperability No load balancing	Simulation
[4]	Optimum cloudlet selection strategy	Latency Bandwidth Response time	Large scaling of cloudlet deployment Optimal cloudlet placement	No workload management No load balancing	Real
[5]	Pareto optimization Bayesian Approach Semi-Markov model	Energy Execution time	Fault recovery system	No power consumption No cost consideration	Simulation
[6]	Hierarchical Stochastic modelling	Time Workload	Workload management	Execution cost is high No load balancing	Simulation
[7]	Resource Intensive Aware Load (RIAL) Balancing	Bandwidth Memory Time cost	Minimize VM communication cost Load balancing	Sub optimal solutions Frequent migrations	Simulation
[8]	Greedy algorithm	Time	Interference control Revenue maximization	No SLA No power consumption Intra-tier interference	Simulation
[9]	Attribute-based security access control techniques	Security Time	Fine-grained mechanism Performance monitoring	Poor QoS No SLA	Simulation
[10]	Practical outsourcing	Cost overhead	Security & efficiency Correctness & soundness	No stability No sparsity Frequent migrations	Simulation
[11]	Encryption of data Integer Linear Programming (ILP)	Power, cost Storage Bandwidth	Security Machine learning based VM allocation	High overhead	Simulation
[12]	Duality Theorem Affine Mapping	Cost	Feasible region protection	No stability	Simulation

Table 14: Heuristics Methods

Paper	Method	Parameter	Highlights	Limitations	Environment
[42]	Ant colony algorithm Greedy-Ant scheduling Forward/backward Dependency	Makespan Execution time Priority	Minimize execution time	No optimal performance Slow	Simulation
[43]	Modified Best-Fit Decreasing (MBFD) with minimization of migrations	Cost Energy	Resource provisioning Autonomic energy-aware mechanism Minimize operational cost	Inefficient workload management No SLA	Simulation
[44]	Elasticity Based Scheduling Heuristic (EBSH) ACO-Honey Bee Optimization	Cost Profit	Random biased sampling Self-managed	Inefficient Slow	Simulation
[45]	Local search	Energy Time Bandwidth	Minimize energy consumption	No load balancing	Real
[46]	Load Balancing based Bayes theorem and Clustering	Cost Makespan Execution time	Maximize posteriori probability value	Low throughput	Simulation

[47]	PSO algorithm	Execution time Cost	Distribution of workload Cost saving	No energy consideration	Simulation
[49]	Critical-Path based heuristic	Execution time Cost	Minimize energy consumption Optimal time management	Low QoS improvement Frequent migrations	Simulation
[52]	Hyper-Heuristic Scheduling Algorithm (HHSA)	Execution time Cost Makespan	Optimization in makespan Reduction in operational cost Cost saving	Inefficient Frequent migrations	Simulation

Table 15:ABC-Based Task Scheduling

Paper	Method	Parameter	Highlights	Limitations	Environment
[53]	Pareto- based ABC	Response time Cost Makespan	High profit Minimize cost Load balancing	No priority Frequent migrations	Simulation
[54]	Power-aware ABC	Power Energy	Energy consumption	Delay No load balancing	Simulation
[56]	Heuristic ABC (HABC)	Makespan Cost	Large Job First (LJF) Maximize resource utilization Load balancing	Inefficient load balancing	Simulation

Table 16:ACO-Based Task Scheduling

Paper	Method	Parameters	Highlights	Limitations	Environment
[57]	Basic ACO	Makespan	Random optimization	Single objective No load balancing Slow	Simulation
[58]	Modified ACO	Response time Throughput	Two level cloud scheduler	High network communication Slow	Simulation
[59]	Load balanced ACO	Makespan	Load balancing	Slower when number of iterations are high	Simulation
[60]	Basic ACO	Energy	Energy aware	No load balancing	Simulation
[61]	Modified ACO	Energy	VM consolidation	No load balancing	Simulation
[62]	Multi objective ACO	Energy Resource usage	Scalability	No load balancing	Simulation
[63]	List ACO	Deadline Cost	Deadline constrained execution cost optimized approach	Slow	Simulation
[64]	LB-ACO	Makespan	Load balancing Multi-objective Scheduling	Sub optimal solutions	Simulation

Table 17:GA-Based Task Scheduling

Paper	Method	Parameter	Highlights	Limitations	Environment
[65]	GA Local Search (LS) technique	Completion Time/Makespan Workload	Minimize completion time	Sub optimal solutions	Simulation
[66]	Johnson's rule based GA	Makespan Cost	Multi-processor scheduling Low complexity	No load balancing	Simulation

Table 18:PSO-Based Task Scheduling

Paper	Method	Parameter	Highlights	Limitations	Environment
[67]	PSO	Makespan Execution time	Optimized execution time	No QoS	Simulation
[68]	Modified PSO GA	Completion time Makespan	Load balancing Minimized Execution time	Slow	Simulation
[69]	MOPSO	Completion time Waiting time Execution time	Minimum time & energy	No load balancing	Simulation
[70]	PSO	Response time Cost	Lower execution time	No scalability	Simulation
[71]	Self- adaptive learning PSO	Makespan Cost	Load balancing based on resource usage	No SLA	Simulation
[72]	PSO	Makespan	Minimizes VMs down time	No SLA	Simulation
[73]	Multi-objective Pareto based PSO	Makespan Cost Energy	Dynamic voltage and frequency scaling	SLA and energy not considered	Simulation
[39]	PSO	Energy	VM placement	No load balancing No SLA	Simulation
[40]	Improved PSO	Energy	VM placement	No load balancing No SLA	Simulation
[41]	Hybrid PSO	Cost Energy	Energy efficient VM placement with PSO-TS	Slow Frequent migrations	Simulation
[74]	PSO for Energy Saving (PS-ES)	Energy Time	Self adaptive Minimize energy	Homogeneous cloud Higher migration rate	Simulation
[110]	Self-Adaptive Learning PSO	Deadline Cost	No formal inter-cloud agreement is need to outsource tasks	No load balancing	Simulation
[111]	Multi-objective PSO	Time Energy	Considered scheduling problem as a discrete task permutation	Only quasi-optimal solutions No load balancing	Simulation
[112]	Heterogeneous dynamic resource provisioning	Deadline Cost	Minimize overall execution cost while meeting a user defined	Convergent time is high Slow	Simulation

			deadline		
[113]	PSO	Cost Time	PSO with embedded cross over and mutation operation	Energy and SLA not considered	Simulation
[114]	PSO	Computation Transmission cost	Simple heuristics PSO with load consideration	Energy and SLA not considered	Simulation
[115]	Discrete PSO	Cost Deadline	Discrete PSO with deadline constraints	No load balancing No SLA	Simulation

Table 19: Hybrid Methods

Paper	Method	Parameter	Highlights	Limitations	Environment
[75]	SA-PSO Temporal delay bound	Delay Cost	Optimized throughput Meet delay bound	No QoS	Simulation
[116]	Hybrid PSO	Makespan Cost Imbalance	List based heuristic algorithm	No SLA	Simulation
[76]	ACO-ABC-PSO Dynamic meta-heuristic	Execution time Makespan Cost Energy	Load balancing Minimize execution time	No QoS	Simulation
[77]	ACO-PSO-SA Scalable multi-objective-Cat Swarm Optimization based SA (CSM-CSOSA)	Execution time Makespan Cost Energy	Load balancing Minimize execution time Reduce operational cost	Slow Frequent migrations	Simulation
[78]	ACO-PSO Hybrid meta-heuristic	Response time Resource utilization	High fault tolerance High resource utilization Low computing time under high load. Low response time.	Homogeneous servers. High cost	Simulation
[79]	Hybrid ACO-PSO	Resource utilization Makespan	Avoids premature solutions	Single objective No load balancing	Simulation
[80]	ACO-PSO with Min-Max	Execution time Cost	Load sharing	Single targeted scheduling No SLA	Simulation
[81]	GA-PSO GA - Hybrid PSO method	Execution time Cost	High resource utilization Low computing time	Low efficiency No SLA	Simulation
[82]	GA-PSO PSOCS-GA	Makespan Cost	Random allocation Resource utilization is high	Slow	Simulation

6. Comments

6.1. Observations

In this survey, we have theoretically reviewed and analysed the issues in resource allocation, task scheduling, VM placement and load balancing techniques proposed in cloud computing. Methods are grouped into different categories based on the objectives, parameters considered, and methodologies used. The highlights, limitations and environment in which experiments were conducted are also briefly analysed. These research outputs are given significant contribution to the computing world to enhance resource management and to provide better QoS to users. Since cloud is a business model, financial considerations is the primary issue to be addressed. Service providers always look for profit and maximum utilization of their resources with minimization of operational cost, energy, while consumer focus on better quality oriented service within minimum cost and time. Based on the review, following comments are made:

QoS consideration: Guaranteeing SLA is the key task in maintaining QoS requirements. Most the works considered time as an important parameter. Heuristic and meta-heuristic approaches consider optimization problems as NP-hard and trying to produce near optimal solutions to acquire QoS requirements. One of the main limitation of these two approaches is the time required to produce the satisfactory allocation is high compared to other methods. But this can be justified with the quality of the results produced by them in the dynamic cloud environment.

Energy conservation: Today green computing is the latest buzz word in the computing industry. Data centres need huge power to run their infrastructure and associated cooling facilities. In order to cool down the temperature due to operation of large server farms, proper air cooling and circulation equipment are installed in datacenters. Server consolidation techniques will reduce the number of servers in the active state, so that power consumption for servers and related cooling equipment can be reduced. Most of the reviewed papers dealing with energy, tries to reduce energy utilization by minimizing the number of physical machines needed to provision user requested VMs.

Optimization methods: In cloud, simultaneous optimizations of all parameters are impossible due to contradictory effect of each one. E.g., time and cost can't be optimized together, since when we try to reduce computation time, it needs powerful servers to complete the task and these powerful machines costs more than slower servers. A multi-objective optimization method gives better solution in this situation. Pareto, heuristic and meta-heuristic methods harness the situation with near optimal solutions to the problem under consideration.

6.2. Open issues

The research works discussed above addressed the major problems in cloud using different techniques. For further enhancement in this field, some unattended issues are to be focused in future. Energy optimization, offers from providers, QoS and SLA considerations are major concerns that need more attention for VM placement and task scheduling in datacenters. Figure 10 shows the pictorial abstract of the open issues in cloud resource and task management.

Resource provisioning: In providing QoS assured services, the different parameters such as energy, delay, time, cost, deadline and profit are to be considered while doing resource provisioning. This research work reveals the issues regarding different resource management techniques and also considers the different allocation schemes for resource provisioning. When demand for the services and users change in real time, there is a need of dynamic resource provisioning methods. The challenges to resource provisioning include dispersion, uncertainty and heterogeneity of resources.

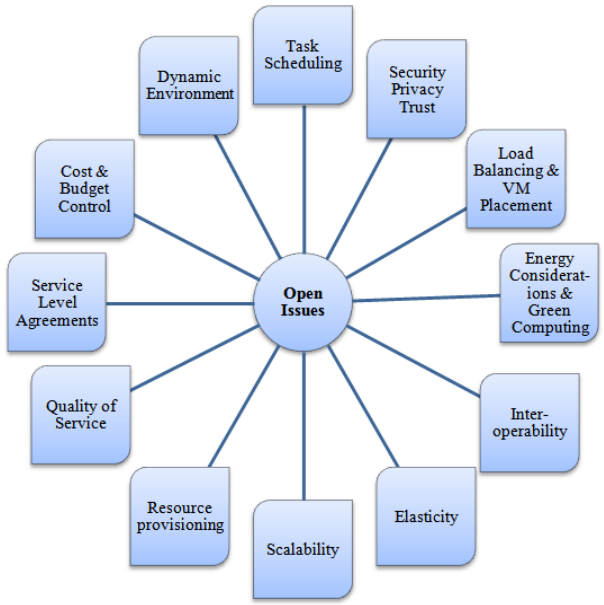

Fig. 10: Open Issues in Cloud.

Task scheduling: To optimize cloud performance, the process of arranging, controlling and optimizing user submitted tasks and workloads are crucial. The proper task scheduling reduces the operational cost and response time. This review discussed various types of scheduling algorithm proposed in cloud. The optimization methods improve the overall efficiency of the scheduling process in terms of time, cost, profit and other performance parameters under consideration. So the task scheduling scheme provide benefits to both customers as well as cloud service providers.

Load balancing and VM placement: The collocation of workload is important in placing it in physical servers. It is the process of allocation of work load across multiple servers for improving the performance of the entire system. This review addressed the problem of imbalance in cloud eco system due to the property of multi-object optimization. This paper also considered different ways to balance the load across servers to minimize execution time as wells as other QoS considerations. The VM placement and live migration are the trendy method to balance the load which is achieved by different heuristic and hybrid algorithms and optimization techniques. The co-allocation can be done by load balancing techniques to get optimal results. Scheduling through load balancing is more suited for cloud environment.

QoS and SLA: Quality of a service depends on customer perception and his Quality of Experience (QoE) about a cloud service. Usually QoE is the difference between perception and expectation about a service. SLAs are important in managing QoS. Intelligent algorithms are needed to keep track user requirements and their expectations.

Scalability: Auto scaling of resources in cloud computing allows resource provisioning dynamically and improves the performance. The scalability of cloud increases the chances to allocate more users and minimize SLA violations. Scalability helps to maintain QoS when the demand of services varies with real time computational environment. The energy, delay, deadline, time and cost affect the scalability and in future these issues are to be addressed in detail for load balancing and VM placement.

Elasticity and inter-operability: Uniform standards are needed to support elastic computing and it should handle related interoperability issues. More sophisticated mechanisms are needed to support integration and interoperability between different cloud providers.

Cost and budget control: In the dynamic cloud, resources and services are being billed per usage, so cost can change an hourly or daily basis. So there is future cost prediction and user information system is needed for transparency. This will aid in budget control for the customers, when they are going for a public cloud.

Energy consideration and green computing: The primary objective of a CSP is to attract more customers and generate more revenue by effective use of their resources. There are several factors contributing to the power usage of a datacenter, including cooling equipments, servers, routers, storage disks, etc. These physical equipmentscan be replaced by high-end systems with low power consumption. But in-order to reduce the energy usage of servers, intelligent power aware resource monitoring and managing methods are needed to support green computing.

Security, privacy and trust: Data security and privacy are the two important issues on the cloud system. The data are scattered in different locations and storage devices in datacenters. The security problems in co-allocation of data arise due to wide variety of applications and users. So counter measures regarding the security issues in cloud computing is important. The research works addressed many problems relating security, trust and privacy concerns. Besides cryptographic methods are proposed to solve the problems in security, new techniques needed to solve more hazards in cloud computing and to increase customer trust.

7. Conclusion

The cloud computing paradigm supports scheduling of tasks in various ways to reduce cost and completion time. Online marketing and purchasing options provides double auction policy to acquire benefits to both the service provider and customer. Heuristic algorithms as well as hybrid algorithms generate near optimal solutions in the dynamic cloud by considering cost and time. Single objective methods produces quick results, because they are only concentrating on one parameter like cost, makespan, delay, energy and profit, but fail to produce accurate results. The allocation of tasks with load balancing results in a low cost and high performance VM Placement. It is one of the best ways of task scheduling for effective results. The survey reveals that, most of the methods proposed used simulation approach than real time implementation. Priority is considered as the main factor to schedule the job to promote the elasticity. Some methods considered homogeneous and heterogeneous cloud to test their methods.

Elastic considerations reduce power consumption and minimize the completion time. SLA-aware task scheduling mainly focuses on agreement between user and provider and which try to reduce SLA violation in the execution process. Mathematical modelling like linear programming models are also able to obtain optimal solution in a reasonable time. Multi-objective task scheduling provides optimal solution with less operational cost, but it is slower in some cases.

Deadline constrained workflow scheduling approach is presented in several papers. They try to provide interrupt free execution of workload within the time and without performance degradation. Nature inspired heuristic algorithms such as ACO, ABC, SA, PSO and GA outperform the scheduling tasks with minimum computation cost and time. But, their performance depends on initial condition and there is a chance to stuck at local optimal solutions. But they produce near optimal solutions in the dynamic cloud environment in a reasonable time. In the case of auction mechanism, online marketing purchasing option gives wonderful option to purchase computational resources for customers. More contributions are needed in this area.

Combination of different methods produces hybrid heuristic task scheduling algorithms. These methods generate comparatively good optimal results due to the hybridization of different heuristic algorithms. But they need more time to converge into an optimal solution. The parameters such as network bandwidth, transfer rate interference and communication cost are less addressed so far.

References

[1] Yi Yao, Jiayin Wang, Bo Sheng, Chiu C. Tan, and Ningfang Mi, “Self-Adjusting Slot Configurations for Homogeneous and Heterogeneous Hadoop Clusters”, IEEE Transactions on Cloud Compu-

ting, Vol. 5, No. 2, pp.344-357, April-June 2017.https://doi.org/10.1109/TCC.2015.2415802.

[2] Yadaiah Balagoni, Rajeswara Rao, "A Cost-effective SLA-Aware Scheduling for Hybrid Cloud Environment", IEEE International Conference on Computational Intelligence and Computing Research,15-17, Dec. 2016, Chennai.https://doi.org/10.1109/ICCIC.2016.7919621.

[3] Kwang Mong Sim, "Agent-based Approaches for Intelligent Intercloud Resource Allocation", IEEE Transactions on Cloud Computing, Volume: PP, Issue: 99.

[4] A Mukherjee, Debashis Deand DG Roy, "A Power and Latency Aware Cloudlet Selection Strategy for Multi-Cloudlet Environment", Volume: PP, Issue: 99, IEEE Transactions on Cloud Computing.

[5] XQiu, Y Dai, Y Xiang, and L Xing, "Correlation Modeling and Resource Optimization for Cloud Service with Fault Recovery", Volume: PP, Issue: 99, IEEE Transactions on Cloud Computing.

[6] Xiaolin Chang, Ruofan Xia, Jogesh K. Muppala, Kishor S. Trivedi, Jiqiang Liu, "Effective Modeling Approach for IaaS Data Center Performance Analysis under Heterogeneous Workload", Volume: PP, Issue: 99, IEEE Transactions on Cloud Computing.

[7] Haiying Shen, "RIAL: Resource Intensity Aware Load Balancing in Clouds", Volume: PP, Issue: 99, IEEE Transactions on Cloud Computing.

[8] Binglai Niu, Yong Zhou, Hamed Shah-Mansouri, and Vincent W. S. Wong, "A Dynamic Resource Sharing Mechanism for Cloud Radio Access Networks", IEEE Transactions on Wireless Communications, Vol. 15, No. 12, pp. 8325 – 8338, December 2016.

[9] Ravi Akella, SaptarshiDebroy, Prasad Calyam, Alex Berryman, Kunpeng Zhu, Mukundan Sridharan, "Security Middle ground for Resource Protection in Measurement Infrastructure-as-a-Service", Volume: PP, Issue: 99, IEEE Transactions on Services Computing.

[10] Cong Wang, Kui Ren, and Jia Wang, "Secure and Practical Outsourcing of Linear Programming in Cloud Computing", INFOCOM, 2011 Proceedings IEEE, 10-15 April 2011, Shanghai, China.

[11] Ali Pahlevan, Xiaoyu Qu, Marina Zapater, David Atienza, "Integrating Heuristic and Machine-Learning Methods for Efficient Virtual Machine Allocation in Data Centers", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems.

[12] Cong Wang, Kui Ren, and Jia Wang, "Secure Optimization Computation Outsourcing in Cloud Computing: A Case Study of Linear Programming", IEEE Transactions on Computers, Volume 65, Issue: 1, pp.216 – 229, Jan. 1 2016. https://doi.org/10.1109/TC.2015.2417542.

[13] Sonia Yassa, Rachid Chelouah, Hubert Kadima and Bertrand Granado, "Multi-Objective Approach for Energy-Aware Workflow Scheduling in Cloud Computing Environments", The Scientific-World Journal Volume 2013, Article ID 350934, 13 pages, https://doi.org/10.1155/2013/350934.

[14] R.K.Jena, "Multi Objective Task Scheduling in Cloud Environment Using Nested PSO Framework", Procedia Computer Science, Vol. 57, 2015, pp. 1219-1227.https://doi.org/10.1016/j.procs.2015.07.419.

[15] Orachun Udomkasemsub, Li Xiaorong, Tiranee Achalakul, "A Multiple-Objective Workflow Scheduling Framework for Cloud Data Analytics", 2012 Ninth International Joint Conference on Computer Science and Software Engineering (JCSSE) 978-1-4673-1921-8/12.

[16] Danlami Gabi, Abdul Samad Ismail, Anazida Zainal, Zalmiyah Zakaria, "Scalability-aware Scheduling Optimization Algorithm for Multi-Objective Cloud Task Scheduling Problem", 2017 6th ICT International Student Project Conference (ICT-ISPC), https://doi.org/10.1109/ICT-ISPC.2017.8075304.

[17] K. Muralitharan, R.Sakthivel, Y.Shi, "Multiobjective optimization technique for demand side management with load balancing approach in smart grid", Journal of Neurocomputing 177(2016) 110–119. https://doi.org/10.1016/j.neucom.2015.11.015.

[18] Heyang Xu, Bo Yang, Weiwei Qi and Emmanuel Ahene, "A Multi-objective Optimization Approach to Workflow Scheduling in Clouds Considering Fault Recovery", KSII Transactions on Internet and Information Systems Vol. 10, No. 3, pp. 976-995, Mar. 2016.

[19] Bahman Keshanchi and Nima Jafari Navimipour, "Priority-Based Task scheduling in the Cloud Systems Using a Memetic Algorithm", Journal of Circuits, Systems, and Computers Vol. 25, No. 10 (2016), World Scientific Publishing Company https://doi.org/10.1142/S021812661650119X.

[20] P. K. Suri, Sunita Rani, "Simulator for Priority based Scheduling of Resources in Cloud Computing", International Journal of Computer Applications,Volume 146 – No.14, pp.10-15, July 2016.

[21] D. I. George Amalarethinam, S Kavitha, "Priority based Performance Improved Algorithm for Meta-task Scheduling in Cloud environment, IEEE 2nd Second International Conference on Computing and Communications Technologies (ICCCT'17) 2017.

[22] Hao Wu, Xiayu Hua, Zheng Li, and Shangping Ren, "Resource and Instance Hour Minimization for Deadline Constrained DAG Applications Using Computer Clouds", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 3, pp. 885–899, March 2016.https://doi.org/10.1109/TPDS.2015.2411257.

[23] Kanchana Viriyapant, Sucha Smanchat, "A Deadline-constrained Scheduling for Dynamic Multi-instances Parameter Sweep Workflow", 15th International Conference on Computer and Information Science (ICIS), IEEE/ACIS, June 26-29, 2016, Okayama, Japan.

[24] Xiaoping Li, Lihua Qian, and Rub´en Ruiz, "Cloud workflow scheduling with deadlines and time slot availability", IEEE Transactions on Services Computing, Volume: PP, Issue: 99.

[25] Mohamed Mohamed, Mourad Amziani, Djamel Belaïd, Samir Tata, Tarek Melliti, "An autonomic approach to manage elasticity of business processes in the Cloud", Future Generation Computer Systems 50 (2015) 49–61.https://doi.org/10.1016/j.future.2014.10.017.

[26] Jiali You, Nannan Qiao, Jinlin Wang, Guoqiang Zhang,Yiqiang Sheng, Haojiang Deng, Xue Liu, "An On-Site Elastic Autonomous Service Network with Efficient Task Assignment", 2016 IEEE 41st Conference on Local Computer Networks Workshops.

[27] Xiaomin Zhu, Ji Wang, Hui Guo, Dakai Zhu, Laurence T. Yang and Ling Liu, "Fault Tolerant Scheduling for Real-Time Scientific Workflows with Elastic Resource Provisioning in Virtualized Clouds", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 12, December 2016, pp.3501-3517.https://doi.org/10.1109/TPDS.2016.2543731.

[28] Yunliang Chen, Lizhe Wang, Xiaodao Chen, Rajiv Ranjan, Albert Y, Zomaya, Yuchen Zhou and Shiyan Hu, "Stochastic Workload Scheduling for Uncoordinated Datacenter Clouds with Multiple QoS Constraints", IEEE Transactions on Cloud Computing, Volume: PP, Issue: 99.

[29] Nikos Tziritas, Samee U. Khan, Thanasis Loukopoulos, Spyros Lalis, Cheng-Zhong Xu, Keqin Li, Albert Y. Zomaya, "Online Inter-Datacenter Service Migrations", IEEE Transactions on Cloud Computing, Volume: PP, Issue: 99.

[30] Fahimeh Ramezani, Jie Lu, Farookh Khadeer Hussain, "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization", International Journal of Parallel Programming, Volume 42 Issue 5, October 2014, pp. 739-754.https://doi.org/10.1007/s10766-013-0275-4.

[31] Hsu-Yang Kung, Ting-HuanKuo, Chi-Hua Chen, Yu-Lun Hsu, "Two-stage cloud service optimisation model for cloud service middleware platform", The Journal of Engineering, Vol. 2018, Iss. 3, pp. 155–161.

[32] C Saravanakumar, C.Arun, "Efficient Idle Virtual Machine Management for Heterogeneous Cloud using Common Deployment Model", KSII Transactions on Internet and Information Systems Vol. 10, No. 4, Apr. 2016.

[33] Aissan Dalvandi, Mohan Gurusamy and Kee Chaing Chua, "Application Scheduling, Placement, and Routing for Power Efficiency in Cloud Data Centers", IEEE Transactions on Parallel and Distributed Systems, Volume: 28, Issue: 4, April 1, 2017, https://doi.org/10.1109/TPDS.2016.2607743.

[34] Shangguang Wang, Zhipiao Liu, Zibin Zheng, Qibo Sun, Fangchun Yang, "Particle Swarm Optimization for Energy-Aware Virtual Machine Placement Optimization in Virtualized Data Centers", 19th IEEE International Conference on Parallel and Distributed Systems, 2013. https://doi.org/10.1109/ICPADS.2013.26.

[35] Konstantinos Tsakalozos, Vasilis Verroios, Mema Roussopoulos, and Alex Delis, "Live VM Migration under Time-Constraints in Share-Nothing IaaS-Clouds", IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 8, August 2017.

[36] Weiwei Kong, Yang Lei, Jing Ma, "Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism", International Journal Optik 127 (2016) 5099–5104, Elsevier. https://doi.org/10.1016/j.ijleo.2016.02.061.

[37] K R Remesh Babu, Philip Samuel, "Virtual Machine Placement for Improved Quality in IaaS Cloud", 2014 IEEE Fourth International Conference on Advances in Computing and Communications, pp.190-194.

[38] Zhifeng Zhong, Kun Chen, Xiaojun Zhai, and Shuange Zhou, "Virtual Machine-Based Task Scheduling Algorithm in a Cloud Computing Environment", Tsinghua Science and Technology ISSN, pp.660-667, Volume 21, Number 6, December 2016.https://doi.org/10.1109/TST.2016.7787008.

[39] Seyed Ebrahim Dashti, Amir Masoud Rahmani, "Dynamic VMs placement for energy efficiency by PSO in cloud computing", Journal of Experimental & Theoretical Artificial Intelligence, Volume 28, 2016 - Issue 1-2: Advances and Applications of Swarm Intelligence, pp.97-112.

[40] Shaobin Zhan, Hongying Huo, "Improved PSO-based Task Scheduling Algorithm in Cloud Computing", Journal of Information & Computational Science 9: 13 (2012) 3821–3829.

[41] Jianen Yan, Hongli Zhang, Haiyan Xu, Zhaoxin Zhang, "Discrete PSO-based workload optimization in virtual machine placement", PersUbiquitComput (2018) 22: 589. https://doi.org/10.1007/s00779-018-1111-z.

[42] Bin Xiang, Bibo Zhang, and Lin Zhang, "Greedy-Ant: Ant Colony System-Inspired Workflow Scheduling for Heterogeneous Computing", IEEE Access, Volume. 5, pp.11404-11412.

[43] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing", Future Generation Computer Systems 28 (2012) 755–768. https://doi.org/10.1016/j.future.2011.04.017.

[44] Ali Al Buhussain, Robson E. De Grande, Azzedine Boukerche, "Elasticity Based Scheduling Heuristic Algorithm for Cloud Environments", 2016 IEEE/ACM 20th International Symposium on Distributed Simulation and Real Time Applications. https://doi.org/10.1109/DS-RT.2016.34.

[45] Yacine Kessaci, Nouredine Melab, El-Ghazali Talbi, "A multi-start local search heuristic for an energy efficient VMs assignment on top of the OpenNebula cloud manager", Future Generation Computer Systems, Volume 36, July 2014, pp, 237-256.

[46] Jia Zhao, Kun Yang, Xiaohui Wei, Yan Ding, Liang Hu, and Gaochao Xu, "A Heuristic Clustering-Based Task Deployment Approach for Load Balancing Using Bayes Theorem in Cloud Environment", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 2, February 2016.https://doi.org/10.1109/TPDS.2015.2402655.

[47] Shengjun Xue, Wenling Shi, Xiaolong Xu, "A Heuristic Scheduling Algorithm based on PSO in the Cloud Computing Environment", International Journal of u- and e- Service, Science and Technology, Vol.9, No. 1 (2016), pp.349-362.

[48] Syed Hamid Hussain Madni, Muhammad ShafieAbd Latiff, Mohammed Abdullahi, Shafi'i Muhammad Abdulhamid, Mohammed Joda Usman, "Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment", PLOS ONE | https://doi.org/10.1371/journal.pone.0176321.

[49] Zhicheng Cai, Xiaoping Li, and Jatinder N.D. Gupta, "Heuristics for Provisioning Services to Workflows in XaaS Clouds", IEEE Transactions on Services Computing, Vol. 9, No. 2, March/April 2016.

[50] Mohammad Masdari, Farbod Salehi1, Marzie Jalali1, Moazam Bidaki, "A Survey of PSO-Based Scheduling Algorithms in Cloud Computing", J NetwSyst Manage, Springer 2016, https://doi.org/10.1007/s10922-016-9385-9.

[51] Mala Kalra, Sarbjeet Singh, "A review of metaheuristic scheduling techniques in cloud computing", Egyptian Informatics Journal (2015) 16, 275–295.https://doi.org/10.1016/j.eij.2015.07.001

[52] Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, and Chu-Sing Yang, "A Hyper-Heuristic Scheduling Algorithm for Cloud", IEEE Transactions on Cloud Computing, Vol. 2, No. 2, April-June 2014.

[53] Asmae Benali, Bouchra El Asri and Houda Kriouile, "A Pareto-based Artificial Bee Colony and Product Line for Optimizing Scheduling of VM on Cloud Computing", 2015 International Conference on Cloud Technologies and Applications (CloudTech).

[54] Kriti Agrawal, Priyanka Tripathi, "Power aware Artificial Bee Colony Virtual Machine Allocation for Private Cloud Systems", 2015 International Conference on Computational Intelligence and Communication Networks.

[55] Elaheh Hallaj, Seyyed Reza Kamel Tabbakh, "Study and Analysis of Task Scheduling Algorithms in Clouds Based on Artificial Bee Colony", Second International Congress on Technology, Communication and Knowledge (ICTCK 2015) November, 11-12, 2015 - Mashhad Branch, Islamic Azad University, Mashhad, Iran.https://doi.org/10.1109/ICTCK.2015.7582644.

[56] Warangkhana Kimpan, Boonhatai Kruekaew, "Heuristic Task Scheduling with Artificial Bee Colony Algorithm for Virtual Machines", Joint 8th International Conference on Soft Computing and Intelligent Systems and 17th International Symposium on Advanced Intelligent Systems, 2016.https://doi.org/10.1109/SCIS-ISIS.2016.0067.

[57] Tawfeek MA, El-Sisi A, Keshk AE, Torkey FA, "Cloud task scheduling based on ant colony optimization", In: 8thintconf computing syst; 2013. p. 64–9. http://dx.doi.org/10.1109/ ICCES.2013.6707172.

[58] Pacini E, Mateos C, Garcı´a C, "Balancing throughput and response time in online scientific clouds via ant colony optimization", AdvEng Software 2015; 84:31–47, Elsevier.https://doi.org/10.1016/j.advengsoft.2015.01.005.

[59] Li K, Xu G, Zhao G, Dong Y, Wang D, "Cloud task scheduling based on load balancing ant colony optimization", Sixth AnnuChinagridConf 2011;2011:3–9. http://dx.doi.org/10.1109/ ChinaGrid.2011.17.

[60] Liu X, Zhan Z, Du K, Chen W, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization", GECCO '14, Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation, pp. 41-48. https://doi.org/10.1145/2576768.2598265.

[61] Ferdaus MH, Murshed M, Calheiros RN, Buyya R, "Virtual machine consolidation in cloud data centers using ACO metaheuristic", In: Euro-Par 2014 parallel process. Springer; 2014. p. 306–17. https://doi.org/10.1007/978-3-319-09873-9.

[62] Y. Gao, H. Guan, Z. Qi, Y. Hou, and L. Liu, "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing", J. Comput. Syst. Sci., vol. 79, no. 8, pp. 1230–1242, 2013.https://doi.org/10.1016/j.jcss.2013.02.004.

[63] Quanwang Wu, Fuyuki Ishikawa, Qingsheng Zhu, Yunni Xia, Junhao Wen, "Deadline-constrained Cost Optimization Approaches for Workflow Scheduling in Clouds", IEEE Transactions on Parallel and Distributed Systems, Volume: PP, Issue: 99, 03 August 2017.

[64] Ashish Gupta, Ritu Garg, "Load Balancing Based Task Scheduling with ACO in Cloud Computing", 2017 IEEE International Conference on Computer an Applications (ICCA), https://doi.org/10.1109/COMAPP.2017.8079781.

[65] Shengxiang Yang, and Sadaf Naseem Jat, "Genetic Algorithms With Guided and Local Search Strategies for University Course Timetabling", IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, Vol. 41, No. 1, January 2011.https://doi.org/10.1109/TSMCC.2010.2049200.

[66] Yonghua Xiong, Suzhen Huang, Min Wu, Jinhua She, and Keyuan Jiang, "A Johnson's-Rule-Based Genetic Algorithm for Two-Stage-Task Scheduling Problem in Data-Centers of Cloud Computing", IEEE Transactions on Cloud Computing.

[67] An-ping Xiong and Chun-xiang Xu, "Energy Efficient Multiresource Allocation of Virtual Machine Based on PSO in Cloud Data Center," Mathematical Problems in Engineering, vol. 2014, Article ID 816518, 8 pages, 2014.

[68] Solmaz Abdi, Seyyed Ahmad Motamedi, and Saeed Sharifian, "Task Scheduling using Modified PSO Algorithm in Cloud Computing Environment", International Conference on Machine Learning, Electrical and Mechanical Engineering (ICMLEME'2014) Jan. 8-9, 2014 Dubai (UAE).

[69] Entisar S. Alkayal, Nicholas R. Jennings, Maysoon F. Abulkhair, "Efficient Task Scheduling Multi-Objective Particle Swarm Optimization in Cloud Computing", IEEE 41st Conference on Local Computer Networks Workshops, 2016.

[70] Dinesh Kumar, Zahid Raza, "A PSO based VM Resource Scheduling Model for Cloud Computing", IEEE International Conference on Computational Intelligence & Communication Technology, 2015.https://doi.org/10.1109/CICT.2015.35.

[71] Liu Z, Wang X, "A PSO-based algorithm for load balancing in virtual machines of cloud computing environment", Lect Notes ComputSci (including SubserLect Notes ArtifIntellLect Notes Bioinformatics) 2012; 7331 LNCS: 142–7.https://doi.org/10.1007/978-3-642-30976-2_17.

[72] Shahrzad Aslanzadeh, Zenon Chaczko, "Load balancing optimization in cloud computing: Applying Endocrine-particale swarm optimization", IEEE International Conference on Electro/Information Technology (EIT), Dekalb, IL, USA, 2015.

[73] Juan J, Durillo, Vlad Nae, Radu Prodan, "Multi-objective energy-efficient workflow scheduling using list-based heuristics", Future Generation Computer Systems, July 2014, Vol.36, pp. 221-236.https://doi.org/10.1016/j.future.2013.07.005.

[74] Zhao, J., Hu, L., Ding, Y., Xu, G., & Hu, M., "A Heuristic Placement Selection of Live Virtual Machine Migration for Energy-Saving in Cloud Computing Environment", PLoS ONE, 9(9), e108275, 2014. https://doi.org/10.1371/journal.pone.0108275.

[75] Haitao Yuan, Jing Bi, Wei Tan, Meng Chu Zhou, Bo Hu Li, and Jianqiang Li, "TTSA: An Effective Scheduling Approach for Delay Bounded Tasks in Hybrid Clouds", IEEE Transactions on Cyber-

netics, Vol. 47, No. 11, November 2017.https://doi.org/10.1109/TCYB.2016.2574766.
[76] Gamal F. Elhady and Medhat A. Tawfeek, "A Comparative Study into Swarm Intelligence Algorithms for Dynamic Tasks Scheduling in Cloud Computing", 2015 IEEE Seventh International Conference on Intelligent Computing and Information Systems (ICICIS'15).https://doi.org/10.1109/IntelCIS.2015.7397246.
[77] Danlami Gabi, Abdul Samad Ismail, "Cloud Scalable Multi-Objective Task Scheduling Algorithm for Cloud Computing Using Cat Swarm Optimization and Simulated Annealing", 2017 8th International Conference on Information Technology (ICIT).
[78] Keng-Mao Cho, Pang-Wei Tsai, Chun-Wei Tsai, Chu-Sing Yang, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing", Neural Computing and Applications, Volume 26 Issue 6, August 2015, pp.1297-1309.https://doi.org/10.1007/s00521-014-1804-9.
[79] Wen X, Huang M, Shi J, "Study on resources scheduling based on ACO algorithm and PSO algorithm in cloud computing", 11th International Symposium on Distributed Computing and Applications to Business, Engineering & Science, 2012, pp. 219–222.
[80] Kanwarpreet Kaur, Amardeep Kaur, "A hybrid approach of load balancing through VMs using ACO, Min Max and genetic algorithm", IEEE International Conference on October 2016, https://doi.org/10.1109/NGCT.2016.7877486.
[81] Sheng-Jun Xue, Wu Wu, "Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm", TELKOMNIKA, Vol.10, No.7, November 2012, pp. 1560-1566.https://doi.org/10.11591/telkomnika.v10i7.1452.
[82] Ali Al-maamari, Fatma A. Omara, "Task Scheduling using Hybrid Algorithm in Cloud Computing Environments", IOSR Journal of Computer Engineering (IOSR-JCE), Volume 17, Issue 3, Ver. VI (May – Jun. 2015), pp. 96-106.
[83] Huandong Wang, Yong Li, Ying Zhang, Depeng Jin, "Virtual Machine Migration Planning in Software-Defined Networks", IEEE Transactions on Cloud Computing.
[84] Walter Cerroni, and Flavio Esposito, "Optimizing Live Migration of Multiple Virtual Machines", IEEE Transactions on Cloud Computing.
[85] Bhaskar Prasad Rimal, and Martin Maier, "Workflow Scheduling in Multi-Tenant Cloud Computing Environments", IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 1, January 2017.https://doi.org/10.1109/TPDS.2016.2556668.
[86] Hamed Shah-Mansouri, Vincent W. S. Wong, and Robert Schober, "Joint Optimal Pricing and Task Scheduling in Mobile Cloud Computing Systems", IEEE Transactions on Wireless Communications, Vol. 16, No. 8, August 2017.https://doi.org/10.1109/TWC.2017.2707084.
[87] Simon S. Woo, Jelena Mirkovic, "Optimal application allocation on multiple public clouds", International Journal Computer Networks 68 (2014) 138–148.
[88] Haitao Yuan, Jing Bi, Wei Tan, and Bo Hu Li, "Temporal Task Scheduling With Constrained Service Delay for Profit Maximization in Hybrid Clouds", IEEE Transactions on Automation Science and Engineering, Vol. 14, No. 1, January 2017.https://doi.org/10.1109/TASE.2016.2526781.
[89] Jincy Joseph, K.R. Remesh Babu, "Scheduling to Minimize Context Switches for Reduced Power Consumption and Delay in the Cloud", 2016 International Conference on Micro-Electronics and Telecommunication Engineering. https://doi.org/10.1109/ICMETE.2016.106.
[90] Xianling Meng, Wei Wang, and Zhaoyang Zhang, "Delay-Constrained Hybrid Computation Offloading with Cloud and Fog Computing", IEEE Access, Volume 5, pp.21355-21367, September 2017. https://doi.org/10.1109/ACCESS.2017.2748140.
[91] Songyun Wang, Zhuzhong Qian, Jiabin Yuan, and Ilsun You, "A DVFS Based Energy-Efficient Tasks Scheduling in a Data Center", IEEE Access, Volume: 5, pp.13090 - 13102, July 2017.
[92] Yibin Li, Min Chen, Wenyun Dai, and Meikang Qiu, "Energy Optimization With Dynamic Task Scheduling Mobile Cloud Computing", IEEE Systems Journal, Vol. 11, No. 1, March 2017.https://doi.org/10.1109/JSYST.2015.2442994.
[93] Hancong Duan, Chao Chen, Geyong Min, Yu Wu, "Energy-Aware Scheduling of Virtual Machines in Heterogeneous Cloud Computing Systems", Future Generation Computer Systems (2016), Volume 74, September 2017, pp. 142-150.
[94] Li Shi, Zhemin Zhang, and Thomas Robertazzi, "Energy-Aware Scheduling of Embarrassingly Parallel Jobs and Resource Allocation in Cloud", IEEE Transactions on Parallel and Distributed Systems, Vol. 28, No. 6, June 2017.https://doi.org/10.1109/TPDS.2016.2625254.
[95] Weiwen Zhang and Yonggang Wen, "Energy-efficient Task Execution for Application as a General Topology in Mobile Cloud Computing", IEEE Transactions on Cloud Computing.
[96] Yaser Mansouri, Adel Nadjaran Toosi, and Rajkumar Buyya, "Cost Optimization for Dynamic Replication and Migration of Data in Cloud Data Centers", IEEE Transactions on Cloud Computing.
[97] Moussa Ehsan, Karthiek Chandrasekaran, Yao Chen, Radu Sion, "Cost-Efficient Tasks and Data Co-Scheduling with Afford Hadoop", IEEE Transactions on Cloud Computing.
[98] Keke Gai, Meikang Qiu, Hui Zhao, "Cost-Aware Multimedia Data Allocation for Heterogeneous Memory Using Genetic Algorithm in Cloud Computing", IEEE Transactions on Cloud Computing.
[99] Sowmya Karunakaran and Rangaraja P. Sundarraj, "Bidding Strategies for Spot Instances in Cloud Computing Markets", IEEE Internet Computing, Volume: 19, Issue: 3, May-June 2015, pp.32 - 40.https://doi.org/10.1109/MIC.2014.87.
[100] Liang Zheng, Carlee Joe-Wong, Chee Wei Tan, Mung Chiang, Xinyu Wang, "How to Bid the Cloud", SIGCOMM '15, Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication, pp.71-84.https://doi.org/10.1145/2785956.2787473.
[101] Maristella Ribs, C.G.Furtado, José Neuman de Souza, Giovanni Cordeiro Barroso, Antão Moura, Alberto S Lima, Flávio R.C Sousa, "A Petri net-based decision-making framework for assessing cloud services adoption: The use of spot instances for cost reduction", Journal of Network and Computer Applications 57 (2015)102–118.https://doi.org/10.1016/j.jnca.2015.07.002.
[102] PeiYun Zhang, and Meng Chu Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy", IEEE Transactions on Automation Science and Engineering, Volume 15, Issue: 2, April 2018.
[103] Lina Ni, Jinquan Zhang, Changjun Jiang, Chungang Yan, and Kan Yu, "Resource Allocation Strategy in Fog Computing Based on Priced Timed Petri Nets", IEEE Internet of Things Journal, Vol. 4, No. 5, pp.772–783, October 2017. https://doi.org/10.1109/JIOT.2017.2709814.
[104] Yi-Li Zhang, Jin-Bai Zhang, "Schedule model in a cloud computing based on credit and cost", Computer Science, Technology and Application, https://doi.org/10.1142/9789813200449_0047.
[105] Neethu B, K.R Remesh Babu, "Dynamic Resource Allocation in Market Oriented Cloud using Auction Method", 2016 International Conference on Micro-Electronics and Telecommunication Engineering, https://doi.org/10.1109/ICMETE.2016.137.
[106] Mohammad Aazam, Eui-Nam Huh, Marc St-Hilaire, Chung-Horng Lung, and Ioannis Lambadaris, "Cloud Customer's Historical Record Based Resource Pricing", IEEE Transactions on Parallel and Distributed Systems, Vol. 27, No. 7, July 2016.https://doi.org/10.1109/TPDS.2015.2473850.
[107] Salah-Eddine Benbrahim, Alejandro Quintero, and Martine Bellaiche, "Live Placement of Interdependent Virtual Machines to Optimize Cloud Service Profits and Penalties on SLAs", IEEE Transactions on Cloud Computing, DOI 10.1109/TCC.2016.2603506,
[108] Parvathy Babu, K.R Remesh Babu, "Cloud Revenue Maximization using Competition and Cooperation", 2016 International Conference on Micro-Electronics and Telecommunication Engineering, https://doi.org/10.1109/ICMETE.2016.138.
[109] Kaiyue Wu, Ping Lu, and Zuqing Zhu, "Distributed Online Scheduling and Routing of Multicast-Oriented Tasks for Profit-Driven Cloud Computing", IEEE Communications Letters, Vol. 20, No. 4, April 2016.https://doi.org/10.1109/LCOMM.2016.2526001.
[110] Xingquan Zuo, Guoxiang Zhang, and Wei Tan, "Self-Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid IaaS Cloud", IEEE Transactions on Automation Science and Engineering, Vol. 11, No. 2, April 2014, pp. 564-573.https://doi.org/10.1109/TASE.2013.2272758.
[111] Hua He, Guangquan Xu, Shanchen Pang, Zenghua Zhao, "AMTS: Adaptive multi-objective task scheduling strategy in cloud computing", China Communications, Year: 2016, Volume: 13, Issue: 4, pp. 162 – 17.
[112] Maria Alejandra Rodriguez, Rajkumar Buyya, "Deadline Based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds", IEEE Transactions on Cloud Computing 2014, Volume 2, Issue 2, pp. 222 – 235.
[113] Lizheng Guo, Shuguang Zhao, Shigen Shen, Changyuan Jiang, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm, Journal of Networks", Vol. 7, No. 3, March 2012, pp.547-553.https://doi.org/10.4304/jnw.7.3.547-553.
[114] Pandey S, Wu L, Guru, Buyya R, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud

computing environments", 24th IEEE International Conference on Advanced Information Networking and Applications 2010.https://doi.org/10.1109/AINA.2010.31.

[115] Zhangjun Wu, Ni Z, Gu L, Liu X, "A revised discrete particle swarm optimization for cloud workflow scheduling", International Conference on Computational Intelligence and Security, IEEE, https://doi.org/10.1109/CIS.2010.46 .

[116] Nazia Anwar and Huifang Deng, "A Hybrid Metaheuristic for Multi-Objective Scientific Workflow Scheduling in a Cloud Environment", Applied Sciences, 8 (2018), 538,.https://doi.org/10.3390/app8040538.

[117] K.R.R. Babu, P. Samuel, Interference aware prediction mechanism for auto scaling in cloud, Computers and Electrical Engineering, Vol. 69(2018) pp. 351-363, https://doi.org/10.1016/j.compeleceng.2017.12.021.

[118] Remesh Babu K.R., Samuel P, "Enhanced Bee Colony Algorithm for Efficient Load Balancing and Scheduling in Cloud", Innovations in Bio-Inspired Computing and Applications. Advances in Intelligent Systems and Computing, vol 424. Springer, Cham.