International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 9, No.2 (Mar-2020)
http://dx.doi.org/10.12785/ijcds/090206

A State-of-Art on Cloud Load Balancing Algorithms

Kaushik Mishra[1] and Santosh Kumar Majhi[2]

[1,2]Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, India

Received 14 Feb.2019, Revised 27 Jan. 2020, Accepted 23 Feb. 2020, Published 01 Mar. 2020

Abstract: The cloud computing is alarmingly getting into mainstream for the booming companies and the research organizations as; they seek to gain benefits from its on-demand access, service models and deployment models. It provides unique features like on-demand access to shared pool of resources over internet in a self-accessible, dynamically scalable and metered manner. It is widely accepted because of its "pay-as-you-go" model. These features make this paradigm a buzzword in the area of high-performance distributed computing (HPDC). Though, this domain is widely accepted still it demands enhancements to bring out the optimized performance. The load balancing among the virtual machines (VMs) belongs to NP-hard problem as far as the equilibrium load distribution is concerned. The hardness of this problem can be defined by considering two factors such as: large solution space and polynomial bounded computation. One of the major issues in cloud computing which, needs serious attention is load balancing for its efficient performance. In the present work, a deep literature study has been carried out by considering the state of art algorithms for cloud load balancing. The algorithm includes traditional methods, heuristic, meta-heuristic, and hybrid approach. From the analysis and study of the methods presented in the deep literature survey, it has been observed that the existing heuristic algorithms are not generating near to optimal solution within polynomial time. The amalgamation of meta-heuristics, and hybrid-heuristics techniques have been proved to produce suboptimal solutions within reasonable time. This paper provides an extensive historical survey and comparative analysis on various existing load balancing (LB) literature. The presented work will be a help hand tool for researchers to design new efficient load balancing algorithms in the Cloud computing domain.

Keywords: Load balancing, Survey of Survey (SoS), Load balancing Algorithms, Heuristic, Meta-heiristic, Hybrid, Cloud Computing

1. Introduction

Computing in the cloud is transfigured into a model encompassing of different services which offers as a utility like electricity, water, telephone etc. To deliver the vision of this utility computing, various computing paradigms such as mainframe, cluster, grid computing have evolved. Due to the outspread of distributed computing everywhere, hence, there is a need to store an enormous bulk of data of an organization and retrieve it efficiently. Then, there is an in need of such a computing platform which not only delivers services but also satisfy customer of various domain at the same time. This model is referred to as *Cloud Computing* in terms of utility. Therefore, this cloud computing can be regarded as a new era of ubiquitous computing which is based on the concept of on-request access to shared pool of assets over web in a self-available, progressively versatile and metered way [1].In this model, user can avail services of clouds based on their requirement despite of their location on a "pay-per-use" basis. This technology is supported by different datacenters putting to use virtualization advancements for amalgamation and compelling usage of resources.

In cloud computing paradigm, consumer avail the services by subscribing to the services they need to use with signing a contract with the cloud vendor called Service Level Agreement (SLA) that defines the Quality of Service (QoS) and parameters under which the services are delivered. This nature of cloud computing is unmistakably communicated by Buyya [1] as *"A cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers"*. It offers the different services as Platform as a Service (PaaS), Infrastructure as a Service (IaaS), Software as a Service (SaaS) and Data as a Service (DaaS) to the customer as per their requirement. This model can be deployed in one of the four ways as Public cloud for general, Private cloud for any solely organization, Hybrid cloud for incorporating the advantages of the aforementioned clouds and Community cloud, explicitly intended to address the necessities of a particular community. The cloud can be called as a service when the following essential characteristics would be

E-mail: kaushikmishra1991@gmail.com, smajhi_cse@vssut.ac.in

incorporated for example, on-request self-service, scalability, expansive system get to, resource pooling, fast elasticity, and estimated service. Hence, it can be termed as ubiquitous computing, as it provides everything-as-a-service.

The popularity of cloud application is so high as the number of consumersincrease;therefore,the demand for loads in terms of number of cores, storage, hardware, software, bandwidth etc.rises greatly. This prompts amplification of efficiency with regard to the utilization of resources by diminishing in the makespan in the cloud. Starvation of resources ought to be limited so as to keep up a compelling service, since this can prompt conceivable overheads [2]. It is in this way, basic to diminish the heap on the server so all clients are given equivalent execution. Resources must be elastic and scalable enough to assimilate the loads among VMs in a cloud environment. This can be overcome by creating the instances of physical machines (PMs) through virtualization in order to deliver the uninterrupted services.It is meant to provide an abstract environment to run any applications by enabling physical machines.

Task allocation to VMs and scheduling them in the load balancing is an NP-hard and NP-complete problem [3]. Tasks are allocated to VMs with respect to some policies and scheduling algorithms. In this context, increasingly viable and productive load balancing algorithms ought to be created to address these issues.

The reminder of this article is structured as follows: in recapitulate, *Section 2* introduces the basic concepts of scheduling and load balancing in cloud computing environment. *Section 3* discusses about the historical survey on load balancing in the literature. *Section 4* presents taxonomy of algorithms used from the advent of cloud computing while highlighting each branches of classification along with the analysis. *Section 5* shows the overall analysis of survey regarding to the proposed taxonomy. *Section* 6 discusses the observations made through our survey. Future directions in addition to the concluding remarks are presented in *Section 7*.

2. BACKGROUND

A. Scheduling on Cloud

The term “Scheduling” is defined by Pinedo in [6] as follows:

“*Scheduling is a decision-making process that is used on a regular basis in many manufacturing and services industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives.*”

The role of a scheduler [5-9] is to find ways to assign loads among nodes evenly to optimize the load balancing objectives by utilizing resources appropriately. In the early days, we had cluster computing that tried to combine one or more standalone systems together in a cluster to work as a single system [10]. It was only using the local resources which were the downfalls of cluster system. This pitfall led to the development of Grid which was integrating heterogeneous systems across the geographically distributed systems [11]. Now-a-days, we have shifted the paradigm of computing from Grid to Cloud to leverage the strengths of both Cluster and Grid [12-15].

There is no such algorithm exist to optimize the computing resources fully because most of the scheduling algorithms are either NP-Complete or NP-Hard [16] despite of seemingly unlimited computing resources in cloud computing environment.

The scheduling problem can be defined as follows: a given set of tasks $T_L = \{T_1, T_2, T_3 \dots T_n\}$ to be assigned to a given set of virtual machines $VM_L = \{VM_1, VM_2 \dots. VM_m\}$ with some predefined parameters. For instance, one of the widely measured parameters is Makespan (MS_{max}) which is defined as the maximum completion time among the tasks and mathematically defined as follows:

$$MS_{max} = \max \{MS_j \,/\, j = 1, 2 \dots n\}, \quad (1)$$

Where, MS is makespan of jobs j.

It is needed to minimize the makespan while ensuring maximum resource utilization in order to achieve the effective load balancing in cloud computing. Fitness function and average utilization are calculated for any load balancing algorithms depending on the specific problem.

The main advantage of scheduling algorithm is to achieve high performance in terms of system throughput by making the use of available resources efficiently. Job scheduling process in cloud can be distributed into three phases; they are Resource discovering and filtering, Resource selection, and Task submission [17]. In **Resource Discovering and Filtering**, datacenter broker discovers the resources present in the system and collects status information from Cloud Information Service (CIS) related to them. During **Resource Selection** process, target resource is selected to submit the task based on certain parameters of task and resource. Then during **Task Submission**, task is submitted to the resource selected.

B. Load Balancing in Cloud

Need of Load Balancing

In a cloud computing environment, the portion of various undertakings to the VM is known as the load. Loads in the cloud system can be categorized as under-loaded or over-loaded or balanced. Load balancing algorithms tries to equalize the total system loads by transferring the workloads from heavily loaded nodes to lightly loaded nodes transparently through cloud migration by aiming to maximize the total system throughput. The load balancing of tasks those are might be dependent or independent on virtual machines (VMs) is a critical part of assignment planning for clouds [18].

Load Balancing QoS Metrics

Some essential load balancing metrics are needed to measure the performance of various load balancing algorithms. Other wisely termed as QoS (Quality of Service) performance metrics in cloud computing environment that effects in load balancing are described below:

- **Throughput:** It indicates that how many processes or user requests (tasks) completed execution per unit time by a virtual machine. High in throughput means greater performance.
- **Response Time:** It is the time duration between the submission and first response of any tasks given to execute in any virtual machine. So, as to achieve a greater performance, this time should be less enough.
- **Makespan:** It is the total time required to complete all the tasks submitted to a virtual machine. This parameter ought to be least, on the grounds that the lesser time algorithm will take for culmination, good will be the performance.
- **Fault tolerance:** It is the capabilities of the system to perform uninterrupted and uniform service even if one or more arbitrary nodes fail.
- **Migration time:** The time required to transfer a task or a virtual machine from one physical machine to another. This migration may be either from a host to another host or a data center to a different data center. To accomplish a good performance in load balancing, this time should be less.
- **Degree of Imbalance:** It measures the imbalance among virtual machines.
- **Energy Consumption:** It is the amount of energy consumed by the devices used in the cloud computing or by the particular data centers.
- **Carbon Emission:** It is the intensity of carbon generated of electric suppliers of a service. Emission of carbon of a service needs to be minimized or controlled though it's badly affecting our environment.
- **Resource utilization:** It is the notch to which the resources of the system like CPU, Memory, Storage, and Networking etc. are uniformly utilized.
- **Reliability:** The task is transferred to any other virtual machine in case of any system failure to enrich the reliability of the system.
- **Band width (BW):** It determines the regulating of outgoing traffic from the local network and incoming traffic sent by an internet agent. This disparity of traffic over a network needs to be managed.

These are the primary metrics to measure the load balancing but apart from these, many researchers have taken many other parameters into consideration as per their requirements such as BW, Overhead, Cost, Accuracy, Predictability, Thrashing, Associated overhead, Reliability, Associated cost.

3. Survey of Survey (SoS)

All articles containing the term "Load Balancing in Cloud Computing" either in the title or in keyword, were first selected from scientific journals including Elsevier, IEEE, Springer, Wiley and other international journals. A tremendous number of works have been done in this topic for cloud computing. We have segregated the articles in terms of survey/review and experimental-based article. This section includes only survey/review article that are further mapped with respect to some parameters and summarized in Table I.

We have presented a comparison of our survey with existing surveys in terms of some parameters that is presented in the following Table II.

4. State-of-Art Algorithms and Analysis

In the literature, the load balancing algorithms were classified either in one of the following ways:

1. Static and Dynamic
2. Based on system state and Based on who initiated the process.
3. Nature inspired and Statistics-based

In this section, we have classified load balancing algorithms into four broad categories depending on their nature of algorithms used for this purpose starting from traditional approach to hybrid heuristics and further sub-divided into categories, as shown in the Fig. 1. Each of these algorithms is briefly discussed in the following subsections. We have presented a review of various algorithms used and implemented by researchers of each category based on some criteria. Hence, we have classified this load balancing algorithm as follows:

A. Traditional Algorithm

This approach is the well-known CPU scheduling algorithms. CPU scheduling is a process which allows one process to be executed while other processes are in waiting queue, thus, to make full use of CPU.

The operating system (OS) selects one of the processes from ready queue for the execution and allocates it to CPU. There are various types of load balancing scheduling techniques exist in distributed computing system. The two major types of traditional algorithms are preemptive and non-preemptive. Preemptive means to preempt the on-going execution to serve a higher priority task and resume the execution once the high priority task is executed. It also depends on a priority which can be internal or external [33]. Each task is assigned with a priority of when a process is going to be executed. Examples of such scheduling algorithms include Round Robin and Priority based scheduling.

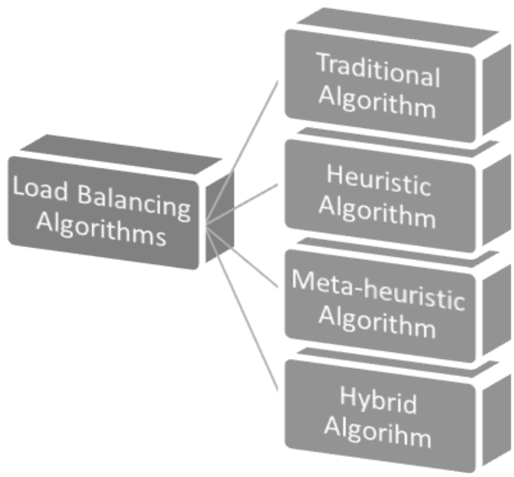

Figure 1. Classification of Load Balancing Algorithms

Non-preemptive scheduling algorithms do not associate with any priority. The task that comes early can able to access the resources. If once the CPU is allocated to a task then the CPU is not taken away until completion. The algorithms that include are FCFS and SJF. CPU maintains a ready queue where all the tasks wait to be executed according to their time of arrival. This queue is generally stored as a linked list.

We have presented a literature review in Table III who has worked on load balancing by using all thesetraditional algorithms.

B. Heuristic Algorithms

A heuristic is regarded as the optimization technique for solving problems more quickly when the classical or traditional techniques are too slow or merely fails to find out the exact solution. Therefore, it is also called as approximation algorithms. The objective of heuristic technique is to produce a solution for a particularproblem in a reasonable time quantum. The solution may not be the best suited but can approximate the optimal solution. This algorithm finds out the possible solution with a tactical guess. It may produce result by themselves or in conjunction with any optimization methods to enhance the performance.

Heuristics techniques can be both static and dynamic. Static heuristic is applied when the completion time of tasks is known in advance. Dynamic heuristic can be used when the arrival of tasks is dynamic in nature. This section focuses on heuristic algorithms [40] such as Min-min, Max-min, RASA (a hybrid approach) and Improved Max-min.

The above discussed heuristic-based load balancing literature is briefly presented in Table IV. It includes the algorithm, research focus, tools used and future scope. From the presented algorithms, it is clear that there are number of contributions towards developing the load balancing algorithms. However, still there is a space for improving the load balancing algorithms.

C. Meta-heuristic Algorithm

The origin of metaheuristics is found in the Artificial Intelligence and Operation Research communities [50]. Heuristic techniques could not able to generate near optimal solution rather it could generate only a very limited number of different solutions. The biggest disadvantage of heuristic methods is to stop at poor quality local optima while finding for the solution which led to the development of an iterative improvement method called as *Metaheuristics* [51-52].Metaheuristics, basically tries to combine the higher-level approximate methods to guide the local improvement procedures in order to effectively and efficiently exploring a search space [53].

According to Voss [54], a meta-heuristic is: *"an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high-quality solutions. It may manipulate a complete (or incomplete) single solution or a collection of solutions per iteration. The subordinate heuristics may be high (or low) level procedures, or a simple local search, or just a construction method."*

In high computational complexity, these exact algorithms cannot be used [55]. Metaheuristics have been successfully applied to real time problems with stringent response time with the advent of increasingly powerful computers and parallel platforms [56-57]. The key idea of this outline is to use three operators as *transition, evaluation and determination*– to search for the possible solution [58]. There are two common transition methods are used as Perturbative and Constructive [59] for the combinatorial problems.

Classification of Metaheuristic Algorithms

Different metaheuristic algorithms which are used in the load balancing can be classified as shown in Fig. 2. The algorithms include Local Search [60], Single Solution, Population based algorithm, evolutionary algorithms and Swarm-based Algorithms [68-69]. In this paper, we have classified the algorithms with respect to the search strategy and the solution-based and categorized the algorithms according to this classification.

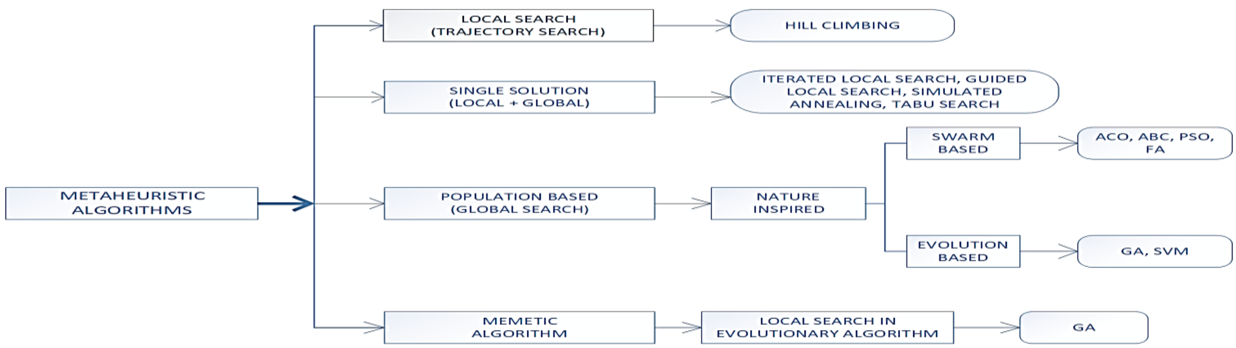

Figure 2. Classification of Meta-heuristic Algorithms

Natural examples of such algorithm include Simulated Annealing [61], Tabu search [62-64], Iterated local search, Guided local search, Genetic programming [65], Evolutionary computation [66], Genetic algorithm [67], Particle Swarm Optimization [70-71], Ant Colony Optimization [72-73], Artificial Bee Colony/Honeybee [74-75], and Intelligent Water Drop (IWD) [76-77], Gravitational Search Algorithm (GSA) [78], BAT algorithm [79], FireFly algorithm [80-81], Cuckoo search [82-83], League Championship Algorithm (LCA) [84] etc.

Summarization of enlisted metaheuristics articles have been presented in the Table V with respect to few factors such as Paper name, Research focus, Compared Algorithm, Simulation environment, Future work etc.

D. Hybrid Algorithm

In this subsection, we have presented a general overview on hybridization. Afterwards, we have analyzed various popular hybrid techniques of load balancing and summarized thereafter. A general classification is also provided that is depicted below in Fig. 3. Several hybrid techniques along with its features are discussed.

In general, Hybrid methods have been proposed along with the metaheuristic techniques to get reaps of the advantages of algorithms while surmounting the pitfalls by integrating the different classes of metaheuristic. Thus, hybrids are believed to benefit from synergy [114]. This mechanism has been popular in the area of optimization by its efficiency of each integrated mechanism. The significance of combining the population-based methods is due to its exploring capability.

We have categorized this hybrid mechanism into three sections by their searching space as shown in the Fig. 3.

Table VI shows the overview of three categories of hybrid classification that have been done in literature.

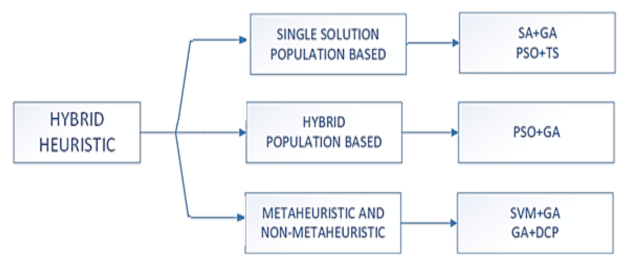

Figure 3. Classification of Hybrid Heuristic Algorithms

5. OVERALL ANALYSIS OF SURVEY

In this section, we have analyzed the survey based on two categorizations as follows:

A. Categorization based on Techniques

We have analyzed only *Metaheuristics Techniques* for this categorization as being depicted in Table VIII and it is evident that during the year 2011 to 2015, most of these techniques were used to solve the load balancing issues in cloud environment. We have also represented this figure through a graph as shown in Fig. 4.

B. Categorization based on Parameters

In this section, we have analyzed the QoS parameters of *Metaheuristic Algorithms* being used in different literature to evaluate the efficiency of proposed load balancing algorithms. It is presented in the Table VII as shown below. We have also represented this analysis through graph as shown in Fig. 5.

6. DISCUSSION

We started out our survey by investigating the different literature and showcased it in Table I. We presented a comparison of our survey with the existing survey in Table II. Furthermore, we have drawn a state-of-the-art algorithm and elaborated each of the branches and summarized the analysis in Table III, IV, V, and VI. At the end, we have presented an overall analysis and survey based on techniques that have been used in solving load balancing problems and parameters, against which they have validated their techniques and conferred it in the Table VII and VIII.

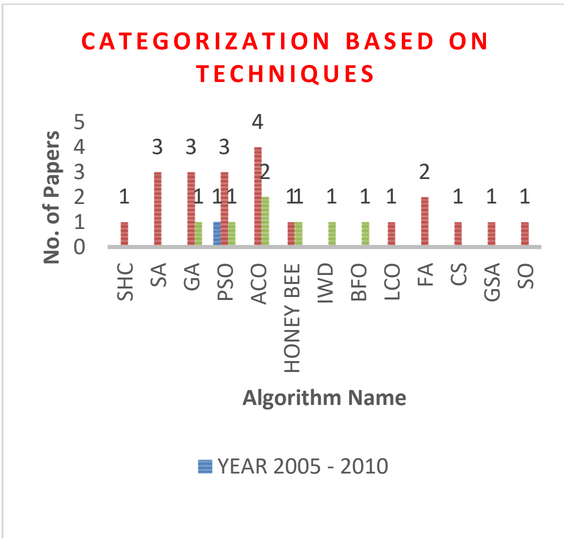

Figure 4. Graphical Representation of Techniques

The following observations have been made based on the survey:

- The traditional algorithms (e.g., FIFO, FCFS, SJF, RR etc.) are rule-based deterministic scheduling algorithms that use only one search direction. These techniques cannot be used for the optimization problem rather can be integrated with some heuristic techniques.
- Heuristic techniques are developed to solve complex problems with low time complexity. It is insufficient to solve when the search space increases with the problem size. It is evident from the literature that, the heuristic methods are incapable to find out the near optimal solution in a reasonable time. Moreover, this algorithm is ineffective to solve large scale multimodal and combinatorial problems.
- Many researchers have applied meta-heuristic algorithms to overcome the drawbacks of heuristic techniques due to its advantages over heuristic. These algorithms are not problem specific; hence, can be employed in many series of problems. They may integrate with other mechanisms to avoid getting stuck in local optima. These algorithms are capable enough to find out near-to-optimal solutions within a reasonable time due to exploration and exploitation. These can be applied in solving combinatorial and multimodal solutions. Literature shows that, these algorithms perform better but they do not guarantee to have an optimal solution due to some unavoidable disadvantages of algorithms. For instance, binary PSO suffers from poor convergence rate and classical PSO has the disadvantage of getting trapped in local optima. GA succumbs to the premature convergence and unpredictable results whereas GSA takes long computational time.In case of GA, it also uses complex parameters in selection and crossover and uses difficult encoding schemes. The quality of solution can be enhanced by generating the initial population. For example, in PSO and GA, the initial population can be obtained by using local search techniques. Many researchers try to improve the quality of solution by modifying the transition operators used in meta-heuristic algorithm [25]. For instance, in ACO, the updating of pheromone greatly affects the search strategy.
  Generally, the amalgamation of two or more metaheuristic algorithms or combing metaheuristic with heuristic reflect the advantages of these algorithms to accomplish the better result in terms of performance and quality of solution. This is termed as Hybrid meta-heuristic.One of the major significances of integration is that the lacunae of one algorithm can be overcome by the advantages and strengths of other algorithms. In case of Single solution, GA is combined with SA and PSO is combined with BF and TS that help in finding the best solution in those local regions.In Population-based hybridization technique, [117] have combined ACO with CS to reap the benefits of both the algorithms while in [118]; ACO is combined with PSO, so that the algorithm should not trap in local optima. In 3rd category, ACO is combined with network theory and GA is combined with the Fuzzy theory. It is witnessed in the literature that it outperforms better than any other algorithms due to its "power of two choices" concept [125].
- The efficiency of each algorithm is measured by its time complexities. From the literature, it is found that time complexity of heuristic approach is $O(lNM^2)$, where l indicates the number of iterations, N is the population size and M is the number of sub solutions. The time complexity of hybrid heuristic is $O(lnm^3)$, where m is the number of machines and n is the number of jobs.

7. Conclusion and Future Scope

The resource constraint environment of cloud computing compels to distribute the loads evenly among the cloud nodes (VMs). During the addressing to the above scenario, the major factors which are considered such as energy consumption and carbon emission along with other QoS criteria at data centers. In our survey, we have presented a unique classification as Survey of Survey (SoS) based on our historical review. In this paper, taxonomy of load balancing algorithms that we have so far is elucidated. Each category of algorithms is further analyzed and summarized the outcomes in the form of tables. This can be extended by optimizing the load distribution amidst several VMs by reducing the aforementioned issues. Hence, there is a great scope of improvement in the underlined algorithms which should be taken as a great concern for further studies.

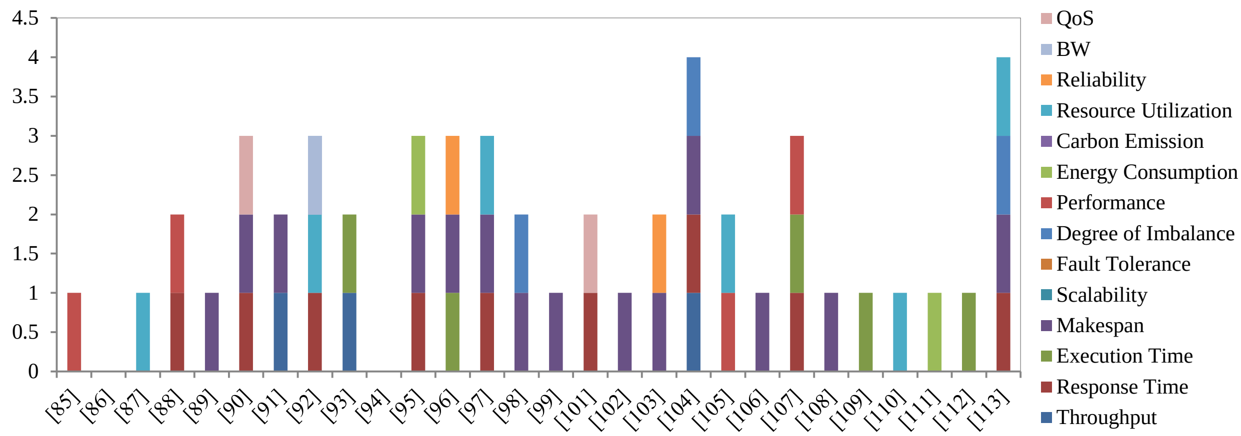

Figure 5. Graphical Representation of QoS Parameters

TABLE I. Survey of Existing Literature Survey

Year	Journal	Author	Title of Paper	Contribution
2010 [19]	IEEE 24th International Conference on Advanced Information Networking and Applications Workshops	Martin Randles, David Lamb, A. Taleb-Bendiab	A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing	• Investigated 3 possible algorithms for load balancing as Honeybee Foraging Behaviour, Biased Random Sampling and Active Clustering. • Simulated the performance result against Throughput vs. System Size and Throughput vs. System Diversity.
2012 [20] [21] [22]	IEEE Second Symposium on Network Cloud Computing and Applications	Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi	A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms	• Classification is based on traditional approach (Static & Dynamic). • Makes comparison based on algorithms they have reviewed and challenges that they have outlined. • One of the algorithms (DDFTP) lacks in terms of storage utilization.
	Journal of Information Systems and Communication	Nidhi Jain Kansal, Inderveer Chana	Existing Load Balancing Techniques in Cloud Computing: A Systematic Review	• Classification based on load balancing techniques. • Makes comparison based on techniques and performance metrics. • Makes an analysis on findings of existing literatures.
	IJCSI International Journal of Computer Science Issues	Nidhi Jain Kansal, Inderveer Chana	Cloud Load Balancing Techniques: A Step Towards Green Computing	• Discussed various load balancing algorithms and presented a comparison based on different parameters (metrics). • Discussed these techniques from the view point of carbon emission and energy consumption.
2013 [23] [24]	International Journal of Computers & Technology	Amandeep Kaur Sidhu, SupriyaKinger	Analysis of Load Balancing Techniques in Cloud Computing	• Classified the algorithms in terms of system load and system topology.
	IJCSI International Journal of Computer Science Issues	Suriya Begum, Dr.Prashanth C.S.R	Review of Load Balancing in Cloud Computing	• Analysedthe 7 load balancing techniques (Event-driven, VectorDot, LBVS, Server-based LB, Fuzzy Logic, PSO, Task Scheduling)
2015 [25] [26]	Egyptian Informatics Journal	Mala Kalra, Sarbjeet Singh	A review of metaheuristic scheduling techniques in cloud computing	• Metaheuristic algorithms (PSO, GA, ACO, LCA & BAT) are surveyed and analysed comparatively for cloud and grid. • Presented a systematic literature review on each algorithm.
	International Journal of Computer Applications	Danlami Gabi, Abdul Samad Ismail, Anazida Zainal	Systematic Review on Existing Load Balancing Techniques in Cloud Computing	• Presented a systematic review on existing load balancing algorithms. • Derived a comparative analysis of all algorithms in terms of their shortcomings, performance metrics, issues addressed.

2016 [27] [28]	Journal ofNetworkandComputerApplications	Jiangtao Zhang, HejiaoHuang, XuanWang	Resource provision algorithms in cloud computing: A survey	• To realize the objectives of resource provisioning, articles and algorithms are surveyed and viewed. • Techniques in algorithms are classified and analysed systematically. • Addressed the issues and lacunae of traditional approaches.
	Journal ofNetworkandComputerApplications	AlirezaSadeghiMilani, NimaJafariNavimipour	Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends	• Proposed a systematic literature review of existing load balancing algorithms (Dynamic & Hybrid). • Presented the properties of different load balancing mechanisms including advantages and disadvantages. • Detailed classification is done based on different cloud metrics. • Addressed the challenges and open issues associated with these algorithms.
2017 [29] [30] [31]	Journal of Network and Computer Applications	EinollahJafarnejad Ghomi, Amir MasoudRahmani, NooruldeenNasih Qader	Load-balancing algorithms in cloud computing: A survey	• Studied and presented a state-of-the-art classification on task scheduling and load balancing algorithms. • Analysed and reviewed 7 categories of load balancing algorithms and summarized it in the QoS metrics. • Provided an insight into the open issues and guidelines for the future research.
	Concurrency Computat: PractExper., Wiley	Minxian Xu, Wenhong Tian, RajkumarBuyya	A survey on load balancing algorithms for virtual machines placement in cloud computing	• Identified the challenges and analysed the existing load balancing algorithms to allocate VM to host in IaaS. • Surveyed algorithms are classified according to the classification. • Provided a comprehensive and comparative analysis of historical load balancing algorithms. • Provided an insight to the researchers for future enhancements.
	Journal of Network and Computer Applications	Avnish Thakur, Major Singh Goraya	A taxonomic survey on load balancing in cloud	• Presented a taxonomic load balancing classification in terms of Nature-inspired and Statistics-based. • Represented each algorithm through flowchart. • Analysed and summarized reviewed state-of-the-art algorithms in tabular manner. • Presented the metrics used in different articles through pie-chart. • Discussed the challenges and open issues along with their possible solution.
2018 [18]	Journal of King Saud University – Computer and Information Sciences	Sambit Kumar Mishra, BibhudattaSahoo, PritiParamitaParida	Load balancing in cloud computing: A big picture	• Outlined a taxonomic survey on load balancing algorithm in terms of Static and Dynamic. • Presented and summarized a detail approach in load balancing algorithms. • Classified the algorithms based on corresponding performance metrics. • Simulated the performance of few heuristic algorithms in terms of Makespan and Energy consumption and represented through graphs.
2019 [32]	Future Generation Computer Systems	AR. Arunarani, D. Manjula, VijayanSugumaran	Task scheduling techniques in cloud computing: A literature survey	• Discussed a comprehensive survey on task scheduling and the associated metrics. • Addressed the various issues pertaining to scheduling and the limitations to overcome. • Distinctive scheduling procedures are studied to discover the usefulness of scheduling characteristics. • Organized literature survey based on 3 measures: methods, application, and parameter-based. • Identified the future research issues related cloud computing.

TABLE II. Comparison of the Present Survey with the Existing Survey

References	[19]	[20]	[21]	[22]	[23]	[24]	[25]	[26]	[27]	[28]	[29]	[30]	[31]	[18]	[32]	
Year	2010		2012		2013		2015		2016			2017		2018	2019	**Present Survey**
Comparative Analysis	Y	Y	Y	Y	N	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
State-of-the-Art	N	N	N	Y	N	N	Y	N	N	Y	Y	N	Y	Y	Y	Y
Graphical Representation	N	N	N	N	N	N	N	Y	N	Y	Y	N	Y	Y	Y	Y
Taxonomy	N	N	N	N	N	N	Y	N	N	N	Y	Y	Y	Y	N	Y
Flowchart Representation	N	N	N	N	N	N	N	N	N	N	N	N	Y	N	N	Y
Survey of Survey (SoS)	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	Y

TABLE III. Literature review of Traditional Algorithms

SL. NO	PAPER NAME	ALGORITHM USED	RESEARCH FOCUS	COMPARED WITH	TOOLS	FUTURE SCOPE
1 [34]	Elastic and flexible deadline constraint load balancing algorithm for cloud computing, 2018	Resource provisioning and de-provisioning	To propose an architecture that capable of handling maximum user requests and elasticity mechanisms using threshold.	FCFS SJF MIN-MIN	CloudSim	NA
2 [MRR] [35]	Task scheduling algorithm in cloud computing based on modified round robin algorithm, 2018	Modified Round Robin (MRR)	To present a review study on various scheduling algorithm with a case study of MRR that shows the average waiting time.	RR	CloudSim	To experiment the effect of other parameters such as VMs, DCs, memory, BW and test it on real environment.
3[HRR] [36]	Round Robin Inspired History Based Load Balancing Using Cloud Computing, 2018	Round Robin Inspire History Based Algorithm	To develop a fog-based environment with the connection of Cloud and Cluster by taking data from end users and by considering two service broker policy.	Honey Bee, RR	Cloud Analyst with Java	Can be extended to simulate on different parameters and with different algorithms.
4 [HEFT] [37]	Modified HEFT Algorithm for task Scheduling in Cloud Environment, 2018.	Modified Heterogenous Earliest Finish Time (HEFT)	To develop a modified algorithm of existing HEFT to distribute the workload among processors in an effective way to reduce the makespan.	HEFT, CPOP	CloudSim and DAG	NA
5 [SJF] [38]	Resource Management and Scheduling in Cloud Computing Environment, 2013	SRTF (SJF)	To propose a model for job-oriented resource scheduling and analyses the algorithms by considering Round Robin, Pre-emptive Priority and Shortest Remaining Time First.	RR, Preemptive Priority (PP)	Traditional approach	To propose a new algorithm and can be compared with other algorithms by taking different parameters.
6 [39]	Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment, 2018	Dynamic Load Balancing Algorithm with Elasticity	To develop a dynamic scheduling algorithm that balances the workload among all the virtual machines with elastic resource provisioning and de-provisioning based on the last optimal k-interval to achieve better scalability.	FCFS SJF MIN-MIN	CloudSim	Can be extended to simulate on different parameters like QoS and can be tested in OpenNebula

TABLE IV. Literature review of Heuristics Algorithms

SL. NO	PAPER DETAILS	ALGORITHM USED	RESEARCH FOCUS	COMPARED WITH	TOOLS	FUTURE SCOPE
1 [41]	QoS guided min-min heuristic for grid task scheduling, 2003	QoS Guided Min-Min heuristic	To propose an algorithm to cater the needs of QoS and deals with many other issues in Grid Computing as security, data administration etc.	Min-Min	GridSim	Can be embedded in multi-dimensional QoS in task scheduling and can be implemented in an actual Grid environment for practical evaluation and theoretical refinement.
2 [42]	A Min-min Max-min Selective Algorithm for Grid Task Scheduling, 2007	Selective Min-Min Max-Min	To develop an algorithm which deals with the cons of two algorithms and take the advantages of both based on standard deviation of the expected completion time of tasks on resources.	Max-Min and Min-Min	GridSim	This is limited to only standard deviation. This selective algorithm can be applied on deadline of each task, cost of execution on each resource, cost of communication and can also be implemented for practical use.
3 [43]	RASA: A New Task Scheduling Algorithm in Grid Environment, 2009	RASA	To develop an algorithm which takes the advantage of and cope up with Max-Min and Min-Min and to estimate the completion time of each task first and schedule under algorithms alternatively.	Max-min, Min-min, OLB, QoS guided Min-min and QoS priority grouping	GridSim	Deadline, arriving rate, cost of each task, execution on each of the resource, cost of the communication and many more can be a topic of research and can be implemented practically on Grid environment.
4 [44]	Improved max-min heuristic model for task scheduling in cloud, 2013	Improved Max-Min	To propose an algorithm to overcome the disadvantages of Max-Min algorithm and to design an algorithm to allocate tasks into resources that build on RASA algorithm.	Min-Min, Max-Min, RASA	Not Mentioned	can be optimized and produced more efficient makespan by using another meta-heuristic approach
5 [45]	User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing, 2013	PA-LBIMM	To propose an improved load balanced algorithm to reduce the makespan and increase the resource utilization (LBIMM) based on Min-Min and also considered user's priority from the perspective of CSP.	Min-Min, LBIMM	Matlab	can be improved this algorithm by considering different parameters like task deadline, QoS, network etc.
6 [46]	Efficient task scheduling algorithms for heterogenous multi-cloud environment, 2015	MCC, MEMAX, CMMN	To propose these algorithms for heterogeneousmulti-cloud environment for scheduling workloads by minimizing makespan and maximizing the cloud utilization.	RR, CLS,CMAXMS,CMMS	MATLAB R2012a	NA
7 [47]	A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment, 2016	Bayes and Clustering (LB-BC)	To propose a heuristic approach to finding the optimal physical host for task deployment by achieving a long-term load balancing strategy for cloud data centres to provide an efficient performance.	RD (Random Deployment), DLB (Optimal Deployment Strategy)	CloudSim	This approach only applies to LAN and can be extended to WAN and also it can be implemented in real world computing.
8 [48]	Task scheduling algorithms for multi-cloud systems: allocation-aware approach, 2017	AXB algorithm (AMinB) (AMaxB) (AMinMaxB)	To propose three allocation-aware task scheduling algorithms for heterogeneous multi-cloud environment to provide a unified service in a collaborative way	Max-Min, Min-Min, Min-Max	MATLAB R2014a	Simulating these algorithms based on different parameters as execution cost, transfer cost and deadline and to design a robust cloud manager by incorporating adverse conditions as slack time, uncertainty and fault tolerance.
9 [49]	Cost Effective Expa-Max-Min Scientific Workflow Allocation and Load Balancing Strategy in Cloud Computing, 2018	Expanded Max-Min (Expa-Max-Min)	To propose an algorithm to effectively give opportunity to the task with maximum and minimum execution time to be scheduled for a reduce cost and time by overcoming Max-Min.	Max-Min, Min-Min	CloudSim	NA

TABLE V. Literature review of Meta-heuristics Algorithms

SL. NO		ALGORITHM USED	PAPER NAME	RESEARCH FOCUS	COMPARED WITH	TOOLS	FUTURE SCOPE
1 [85]	LOCAL SEARCH	Stochastic Hill Climbing (SHC)	Load balancing in Cloud Computing using Stochastic hill Climbing - A Soft Computing Approach, 2012	To propose a soft computing approach to allocate incoming jobs to VMs.	RR, FCFS	Cloud Analyst	This problem can be tested and implemented using other soft computing approaches for better improvement.
2 [86]	SINGLE SOLUTION	Simulated Annealing (SA)	Simulated-Annealing Load Balancing for Resource Allocation in Cloud Environments, 2013	To present a simulated-annealing load balancing algorithm for solving the resource allocation and scheduling problem in a cloud computing environment.	RR, Basic SA	Cloud Sim	Multi-dimensional resources need to be considered and migration cost as a performance metrics should be taken into consideration.
3 [87]			Resource Allocation in Cloud using Simulated Annealing, 2014	To propose a method (Priority Fit SA) to solve the resource allocation problem in multi-layers cloud computing.	FCFS	JAVA Simulator	Cooling speed may vary with respect to temperature.
4 [88]			Simulated Annealing (SA) based Load Balancing Strategy for Cloud Computing, 2015	To propose a method that balances the loads among the VMs in cloud environment.	RR, FCFS, SHC	Cloud Analyst	This problem can be tested and implemented using other soft computing approaches for better result.
5 [89]	POPULATION BASED (EVOLUTIONARY)	Genetic Algorithm (GA)	GA-based task scheduler for the cloud computing systems, 2010	This paper proposes a new scheduler which makes a scheduling decision by evaluating the entire group of tasks in the job queue. GA is used as optimization method for the new scheduler.	FIFO, Delay Scheduling	JAVA Simulator	The balance of GA computation time and efficiency need to be done. More focus needs to pay for predicting the execution time of each task.
6 [90]			A Genetic Algorithm (GA) based load balancing strategy for cloud computing, 2013	To balance the load of the cloud infrastructure by minimizing makespan of a given tasks by considering a scenario of internet banking of an International Bank.	RR, FCFS, SHC (own proposed approach)	CloudSim, Cloud Analyst	Variation of crossover and selection strategies of GA could be applied for getting a more efficient and tuned result.
7 [91]			Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing, 2014	To develop an algorithm that works not only for shorter and average jobs' makespan but also to balance the inter-nodes loads. Based on double-fitness function (Spanning time & load balancing (JLGA)) and adopts greedy algorithm to initialize the populations.	Adaptive Genetic Algorithm (AGA)	Matlab	Jobs' priority must be taken into consideration and to adopt dynamic global adaptive control strategy in genetic algorithm.
8 [92]			Effective Resource utilization in a cloud computing environment through a dynamic well-organized load balancing algorithm for VMs, 2016	To develop a well-organized load balancing algorithm in order to distribute the loads on VMs in the network by reducing the cost and energy consumption and avoiding the congestion in the network by improving the performance.	Not Mentioned	VMWare workstation 9VMotion)	Servers may be added to testify the results.
9 [93]		Particle Swarm Optimization (PSO)	A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments, 2010	To present a particle swarm optimization (PSO) based heuristic to schedule applications to cloud resources that takes into account both computation cost and data transmission cost and it outperforms better than compared one.	Best Resource Selection (BRS) Algorithm	JSwarm	Can integrate PSO based heuristic into our workflow management system to schedule workflows of real applications such as brain imaging analysis, EMO, and others.

10 [94]	POPULATION BASED (SWARM-INSPIRED ALGORITHM)		A PSO-Based Algorithm for Load Balancing in Virtual Machines of Cloud Computing Environment, 2012	Improved the standard PSO and introduce a simple mutation mechanism and a self-adapting inertia weight method by classifying the fitness values. Optimizes the execution time in terms of running time and resource utilization.	SPSO	Matlab	The problems like restrictions from bandwidth, problems in job decomposition, and energy costs of cloud data centres need to be addressed.
11 [95]			Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization (TBSLBPSO), 2014	To propose a TBSLBPSO that achieves system load balancing by only transferring extra tasks from an overloaded VM instead of migrating the entire overloaded VM to a new VM and to develop a multi-objective PSO (MOPSO) to migrate the loads.	Not Specific. (other traditional methods for load balancing)	CloudSim, Jswarm	Can be extended by integrating MOPSO in Jswarm and considers other aspects of task scheduling optimization as objective.
12 [96]			Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments, 2015	To propose a model for scheduling and allocating tasks to VMs by considering different parameters such as reliability, execution time, transmission time, make span, round trip time, transmission cost.	standard PSO, random algorithm, LCFP	Not mentioned	Can be tested for other parameters of load balancing and cam be used for any number of tasks and resources.
13 [97]			A PSO based task scheduling algorithm improved using a load balancing technique for cloud computing environment, 2017	To present a static task scheduling method based on PSO where tasks are assumed to be non-pre-emptive and independent by improving the performance of basic PSO	RR, Improved PSO, Improved RASA	CloudSim, ANOVA	Can be expanded the method for workflow application and taking QoS criteria like fault tolerance capability and cost reduction
14 [98]		Ant Colony Optimization (ACO)	Cloud task scheduling based on load balancing ant colony optimization, 2011	To propose an algorithm (LBACO) that aims to minimize the makespan of a given task sets.	FCFS, Basic ACO	CloudSim	Can be implemented in real computing, availability vector should be considered for heterogeneous processing.
15 [99]			Ant Colony Optimization: A solution of load balancing in cloud, 2012	To develop an effective load balancing algorithm using ACO to maximize and minimize different parameters for the clouds of different sizes, propose to develop an ant-based control system to implement two sets of mobile agents.	Not Mentioned	NS2	Gap: no formation of cluster in the cloud and does not consider fault-tolerance issue. Future Work: implementation this in a complete cloud environment
16 [100]			Load balancing of nodes in cloud using ant colony optimization, 2012	To propose a modified version of standard ACO to distribute the loads among VMs where the ants continuously update a single result set rather than updating their own result set.	Basic ACO	JAVA Simulator	Can be compared with any other algorithm for performance and can be validate using the performance parameters.
17 [101]			An ant colony based load balancing strategy in cloud computing, 2014	Proposes a novel ant colony-based algorithm to balance the load by searching under loaded node	FCFS, SHC, GA, existing ACO	Cloud Analyst	Can proceed to include the fault tolerance and different function variation to calculate the pheromone value can be used for further research work.
18 [102]			Dynamic and Elasticity ACO load balancing algorithm for cloud computing, 2017	To present an innovative, dynamic and elastic algorithm to perform load balancing among existing systems in a data centre by using ACO and makespan as a parameter	FCFS, M-ACO, ACO	CloudSim	Can be extended and tested against other parameters for the reliability and scalability terms
19 [103]			Load Balancing based task scheduling with ACO in cloud computing, 2017	To propose a meta-heuristic approach of ACO algorithm to solve the task scheduling problem in cloud environment by minimizing the makespan/computation time to have better load balancing.	NAGA-ii algorithm	CloudSim	Can be extended by taking other parameters into consideration and compare with other existing algorithms to have better performance.

20 [104]		Honey Bee	Honey Bee behavior inspired load balancing of tasks in cloud computing environment, 2013	To achieve well balanced load across VMs for maximizing throughput and to balance the priorities of tasks on the machine to achieve minimal waiting time.	DLB, FIFO, WRR	CloudSim	Extend the load balancing for workflows with dependent tasks; plan to improve the QoS parameter.
21 [105]			Load balanced Transaction Scheduling Using HoneyBeeoptimization Considering Per formability in On-Demand computing system, 2017	To enhance the resource availability by decreasing the load and also increases per formability and reduces miss ratio.	ACO, HLBA, DLB and Randomized	Soft Computing approach and Colored Petri Nets	Can be extended this work to analyse the dependability of the system
22 [106]		Intelligent Water Drops (IWD)	An Extended Intelligent Water drops Algorithm for workflow scheduling in cloud computing environment.	To propose an extended algorithm of the natural-based Intelligent Water Drops (IWD) for scheduling on a set of workflows of different types and sizes in cloud to have a greater performance.	Min-Min, MaxMin, FCFS, RR, MCT, PSO, C-PSO	Workflow simulator	Plan to replace the proposed IWD with an Improved IWD algorithm by considering other parameters such as energy usage of the resources.
23 [107]		Bacteria Foraging Optimization (BFO)	Online and Offline based load balance algorithm in cloud computing, 2017	To propose an improved algorithm where online algorithm is used to deal with the load balance problems and offline algorithm is used for global scheduling to ensure the stability of whole systems.	Greedy, RR, Min-Min, MinF, MaxMin, MaxF, BFO, BFO-H	CloudSim	To improve the performance of different cloud scheduling systems based on proper choice of load balancing algorithm
24 [108]		League Championship Algorithm [LCA]	Tasks Scheduling Technique using League Championship Algorithm for Makespan Minimization in IAAS Cloud, 2014	Propose a League Championship Algorithm (LCA) based makespan time minimization scheduling technique in IaaS cloud.	FCFS, Last Job First (LJF) and Best Effort First (BEF)	Matlab	This technique can be used and tested for other metrics and can be applied in various domains like big data, routing problem in distributed networks, learning the Structure of Bayesian networks, assignment problem in graph coloring and other known NP-hard problems.
25 [109]		FireFly Algorithm (FA)	A Fuzzy-based Firefly Algorithm for Dynamic Load Balancing in Cloud Computing Environment, 2014	To propose an algorithm that proficiently balances the loads and efficiently utilizes the resources. Develop an algorithm for the cloud in a partitioned cloud environment to balance the load across the variety of partitions.	GA	JAVA Simulator	A monitoring and analysing mechanism is necessary for further optimization in load balancing strategies.
26 [110]			A Load Balancing Model Using Firefly Algorithm in Cloud Computing, 2014	To propose a model that can overcome the exploration and exploitation problem of traditional GA and optimize the performance of load balancing	Improved AGA	CloudSim	Can be tested by considering other performance metrics and compare this algorithm with efficient and optimized load balancing techniques.
27 [111]		Cuckoo Search (CS)	Proposing a Load Balancing Method Based on Cuckoo Optimization Algorithm for Energy Management in Cloud Computing Infrastructures, 2015	To present an approach based on Cuckoo Optimization Algorithm (COA) to detect over-utilized hosts. Employs the Minimum Migration Time (MMT) policy to migrate VMs from the over-utilized hosts to the other hosts.	MAD-MMT, IQR-MMT, Bee-MMT, LR-MMT	CloudSim	Can propose a much effective underutilized method to manage under-loaded hosts; following that we should consider other metrics like response time as a requirement factor to guaranty a high quality of service (QOS) to satisfy the customers.
28 [112]		Gravitational Search Algorithm (GSA)	Cloudy GSA for load scheduling in cloud computing, 2013	Proposes a near optimal load scheduling algorithm named Cloudy-GSA to minimize the transfer time and the total cost incurred in scheduling the cloudlets to the VMs and achieved by fitness value.	Segmented Min-Min, GSA,SA, GA,Tabu Search, Min-Min,	Network CloudSim	Aims to minimize the total cost by using improved fitness function considering other parameters for better minimized results in the cloud computing environment, based on swarm intelligence to further reduce the total cost.

					FCFS, PSO		
29 [113]	**MIMIC**	Symbiotic Organism (SO)	Symbiotic organism search based task scheduling in cloud computing environment, 2015	To develop a metaheuristic optimization technique called DSOS which should perform significantly better than PSO for a large search space.	PSO	CloudSim	Can be compared with other discrete optimization problem, a multi objective SOS can be designed by taking other parameters when scheduling tasks.

TABLE VI. Literature review of Hybrid Algorithm

SL. NO		PAPER NAME	ALGORITHM USED	RESEARCH FOCUS	COMPARED WITH	TOOLS	METRICS	FUTURE SCOPE
1 [115]	**Single Solution**	Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment, 2010	GA+SA	The genetic simulated annealing algorithm considers the QOS requirements of different type tasks, corresponding to the characteristics of user tasks in cloud computing environment.	No Algorithm	Not mentioned	QoS (CT, BW, cost, distance, reliability)	Not Mentioned
2 [116]		An Enhanced Task Scheduling Algorithm on Cloud Computing Environment, 2016	PSO+BF+TS (BFPSOTS)	To introduce and implement a hybrid algorithm to assign the users' tasks to multiple computing resources. The aim is to reduce the execution time, and cost, as well as, increase resource utilization.	Standard PSO	CloudSim	Makespan, Cost, Resource Utilization	Plan to improve PSO algorithm using other greedy algorithms (e.g., Worst-Fit), and considering dependent tasks instead of independent tasks to achieve the high performance of the overall system.
3 [117]	**Population-based Hybrid**	Minimizing the Makespan using Hybrid Algorithm for Cloud Computing, 2013	ACO+CS	To propose a hybrid algorithm this combines the advantage of ACO and Cuckoo search. The makespan can be reduced with the help of hybrid algorithm, since the jobs have been executed with in the specified time interval by allocation of required resources.	ACO	Cloud Lab	Makespan	Can be utilized for more and more jobs.
4 [118]		A hybrid meta-heuristicalgorithm forVM scheduling with load balancing in cloud computing, 2015	ACO+PSO (ACOPS)	To propose an algorithm that uses historical information to predict the workload of new input requests to adapt to dynamic environments without additional task information.	ACO, PRACOSA, GA, ACS FCFS+RR	cloud system with star topology, C	Makespan, DOI, Performance	Considering VM scheduling to be a multi-objective problem and developing a multi-objective optimization algorithm to solve it
5 [119]		A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing, 2018	CSO + IRRO	To propose a framework ICDSF that combines advantages of these two algorithms for scheduling of tasks in cloud. Performance is measured against the benchmark function (CEC 2017), this hybrid algorithm is tested using the NASA-iPSC. Results show incredibly well.	IRRO, RRO, CSO, BAT, PSO	CloudSim, Matlab	Makespan, Response Time, Throughput	Fuzzy logic could be used for the segregating of ravens, CSO can be combined with other evolutionary algorithms, IRRO can be enhanced by compounding with EFO, SA, applying some operators (mutation) from evolutionary algorithms might be helpful.
6 [120]		Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm, 2018	GA+BF	To propose a hybrid algorithm that aims to minimize the makespan and reduces the energy consumption. This algorithm outperforms better in terms of convergence, stability and solution diversity.	PSO, GA, BFA	Matlab R2013a	Makespan, Energy Consumption	Algorithms' performance can be improved, exploration of operators and parameters are needed, each factor related to the similarity function needs to be examined.

7 [121]		Binary PSOGSA for load balancing task scheduling in cloud environment, 2018	Bin-LB-PSOGSA	To propose a binary load balancing algorithm using a hybrid PSO and GSA to find the best tasks-to-VMS mapping efficiently to enable the scheduling process by improving load balance level.	Pure-Bin-LB-PSO	CloudSim	Makespan, Cost, Flowtime	Plan to extend this kind of load balancing for workloads with different tasks by considering other QoS parameters
8 [122]	**Metaheuristic + Non-Metaheuristic**	A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation, 2010	ACO + Network Theory	To propose a load balancing mechanism (ACCLB) based on ant colony and complex network theory in open cloud computing federation	Search Min Mechanism (SMM), Classic Ant Colony (CAC)	JAVA	Standard Deviation	How the parameters α, β and γ in this literature effect each other will be further studied in future.
9 [123]		Hybrid Job scheduling Algorithm for Cloud computing Environment, 2014	GA + Fuzzy Theory	To modify the standard Genetic algorithm and to reduce the iteration of creating population with the aid of fuzzy theory. The main goal of this research is to assign the jobs to the resources with considering the VM MIPS and length of jobs.	ACO, MACO	CloudSim	ET(Makespan), EC, DOI	Not Mentioned
10 [124]		FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method, 2015	GA + Fuzzy Theory	Present a hybrid fuzzy-based steady-state GA approach called FUGE that is based on fuzzy theory and a genetic algorithm (GA) that aims to perform optimal load balancing considering execution time and cost.	ACO, MACO	CloudSim	Execution Time (Makespan), Execution Cost, Degree of Imbalance (DOI)	Can be considered energy consumption and VM migration to make efficient, consistence power saving model, can be further optimize the calculation of the fitness value and crossover steps by considering the VM energy consumption as input parameters of the fuzzy system.

TABLE VII. Categorization based on Parameters

Reference	Throughput	Response Time	Execution Time	Makespan	Scalability	Fault Tolerance	Degree of Imbalance	Performance	Energy Consumption	Carbon Emission	Resource Utilization	Reliability	BW	QoS	Others
[85]								Y							
[86]															Standard deviation, Iteration number, The influence of parameter (α)
[87]											Y				
[88]		Y						Y							
[89]				Y											
[90]		Y		Y										Y	
[91]	Y			Y											
[92]		Y									Y		Y		
[93]	Y		Y												Total cost of Computation and Total cost of Execution.
[94]															Iteration number
[95]		Y		Y					Y						
[96]			Y	Y								Y			Availability, Round-trip time, Transmission time & Cost
[97]		Y		Y							Y				
[98]				Y			Y								

[99]				Y											CPU load, Memory Capacity, delay or network load.
[101]		Y												Y	
[102]				Y											
[103]				Y								Y			
[104]	Y	Y		Y			Y								
[105]								Y			Y				Availability
[106]				Y											
[107]		Y	Y					Y							
[108]				Y											
[109]			Y												computational time, load arrived, task migration and the cost
[110]											Y				CPU utility rate, Memory usage rate
[111]									Y						
[112]			Y												Transfer Time, Total Cost of Execution Time
[113]		Y		Y			Y				Y				CPU utilization

TABLE VIII. Categorization based on Techniques

LOAD BALANCING TECHNIQUES	YEAR		
	2005 – 2010	**2011 – 2015**	**2016 - 2019**
SHC		[85]	
SA		[86] [87] [88]	
GA		[89] [90] [91]	[92]
PSO	[93]	[94] [95] [96]	[97]
ACO		[98] [99] [100] [101]	[102] [103]
HONEY BEE		[104]	[105]
IWD			[106]
BFO			[107]
LCO		[108]	
FA		[109] [110]	
CS		[111]	
GSA		[112]	
SO		[113]	

Acknowledgment

The authors are very much thankful to the anonymous reviewers and editors for their constructive suggestions which significantly facilitate the quality improvement of the article.

References

[1] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility,” Future Generation computer systems, vol. *25*(6), pp. 599-616, 2009.

[2] R. Buyya, C. vecchiola and S. T. Selvi, Mastering cloud computing: foundation and application programming, Mc Graw Hill Education, Newnes, 2013.

[3] E. Kumari and Monika, “A review on task scheduling algorithms in cloud computing,” International Journal of Science, Environment and Technology, vol. 4, no. 2, pp. 433 – 439, 2015.

[4] S. Kumar, and R. H. Goudar, “Cloud computing-research issues, challenges, architecture, platforms and applications: A survey,” International Journal of Future Computer and Communication, vol. 1(4), 356, 2014.

[5] R. Nallakumar, N. Sengottaiyan, and S. Nithya, “A survey of task scheduling methods in cloud computing,” International Journal of Computer Sciences and Engineering, vol. *2*(10), pp. 9-13, 2014.

[6] M. Pinedo, Scheduling: Theory, Algorithms and Systems. Englewood Cliffs, NJ, USA: Prentice-Hall, Inc., 1995.

[7] P. Chr´etienne, E. G. Coffman, J. K. Lenstra, and Z. Liu, Eds., Scheduling Theory and Its Applications. New York: Wiley, 1995.

[8] A. Allahverdi, C. T. Ng, T. C. E. Cheng, and M. Y. Kovalyov, “A survey of scheduling problems with setup times or costs,” Eur. J. Oper. Res., vol. 187, no. 3, pp. 985–1032, 2008.

[9] C. N. Potts and V. A. Strusevich, “Fifty years of scheduling: A survey of milestones,” J. Oper. Res. Soc., vol. 60, no. S1, pp. S41–S68, 2009.

[10] G. F. Pfister, In Search of Clusters, 2nd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 1998.

[11] I. Foster, “The grid: A new infrastructure for 21st century science,” in Grid Computing: Making the Global Infrastructure a Reality. New York: Wiley, 2003, pp. 51–63.

[12] I. Foster, Y. Zhao, I. Raicu, and S. Lu, “Cloud computing and grid computing 360-degree compared,” in Proc. Grid Comput. Workshop, 2008, pp. 1–10.

[13] B. P. Rimal, E. Choi, and I. Lumb, “A taxonomy and survey of cloud computing systems,” in Proc. Int. Joint Conf. INC, IMS IDC, pp. 44–51, 2009.

[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," Dept. Elect. Eng. Comput. Sci., Univ. Calif., Berkeley, Tech. Rep. UCB/EECS- 2009-28, 2009.

[15] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A view of cloud computing," Commun. ACM, vol. 53, no. 4, pp. 50–58, 2010.

[16] M. R. Garey and D. S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness. New York: Freeman, 1979.

[17] S. C. Wang, K. Q. Yan, S. S. Wang, and C. W. Chen, "A Three-Phases Scheduling in a Hierarchical Cloud Computing Network," IEEE, 2011.

[18] S. K. Mishra, B. Sahoo, and P. P. Parida, "Load balancing in cloud computing: A big picture," Journal of King Saud University-Computer and Information Sciences, 2018.

[19] M. Randles, D. Lamb, and A. Taleb-Bendiab, "A comparative study into distributed load balancing algorithms for cloud computing," 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 551-556, IEEE, April, 2010.

[20] K. Al Nuaimi, N. Mohamed, M. Al Nuaimi, and J. Al-Jaroodi, "A survey of load balancing in cloud computing: Challenges and algorithms," Second Symposium on Network Cloud Computing and Applications (NCCA), pp. 137-142, IEEE, December, 2012.

[21] N. J. Kansal, and I. Chana, "Existing load balancing techniques in cloud computing: A systematic review," Journal of Information Systems and Communication, vol. 3(1), 87, 2012.

[22] N. J. Kansal, and I. Chana, "Cloud load balancing techniques: A step towards green computing," IJCSI International Journal of Computer Science Issues, vol. 9(1), pp. 238-246, 2012.

[23] A. K. Sidhu, and S. Kinger, "Analysis of load balancing techniques in cloud computing," International Journal of computers & technology, vol. 4(2), pp. 737-741, 2013.

[24] S. Begum, and C. S. R. Prashanth, "Review of load balancing in cloud computing," International Journal of Computer Science Issues (IJCSI), vol. 10(1), 343, 2013.

[25] M. Kalra, and S. Singh, "A review of metaheuristic scheduling techniques in cloud computing," Egyptian informatics journal, vol. 16(3), pp. 275-295, 2015.

[26] D. Gabi, A. S. Ismail, and A. Zainal, "Systematic review on existing load balancing techniques in cloud computing," International Journal of Computer Applications, vol. 125(9), pp. 16-24, 2015.

[27] J. Zhang, H. Huang, and X. Wang, "Resource provision algorithms in cloud computing: A survey," Journal of Network and Computer Applications, vol. 64, pp. 23-42, 2016.

[28] A. S. Milani, and N. J. Navimipour, "Load balancing mechanisms and techniques in the cloud environments: Systematic literature review and future trends," Journal of Network and Computer Applications, vol. 71, pp. 86-98, 2016.

[29] E. J. Ghomi, A. M. Rahmani, and N. N. Qader, "Load-balancing algorithms in cloud computing: a survey," Journal of Network and Computer Applications, vol. 88, pp. 50-71, 2017.

[30] M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," Concurrency and Computation: Practice and Experience, vol. 29(12), e4123, 2017.

[31] A. Thakur, and M. S. Goraya, "A taxonomic survey on load balancing in cloud," Journal of Network and Computer Applications, vol. 98, pp. 43-57, 2017.

[32] A. R. Arunarani, D. Manjula, and V. Sugumaran, "Task scheduling techniques in cloud computing: A literature survey," Future Generation Computer Systems, vol. 91, pp. 407-415, 2019.

[33] **V.** Rajeshram, and S. Rajasulochana, "Scheduling Algorithms: A Landscape on Pros and Cons," International Journal of Engineering, Research and Technology (IJERT), vol. 3, issue 12, 2014.

[34] M. Kumar, K. Dubey, and S. C. Sharma, "Elastic and flexible deadline constraint load Balancing algorithm for Cloud Computing," Procedia Computer Science, vol. 125, pp. 717-724, 2018.

[35] R. A. Khurma, H. E. B. A. Al Harahsheh, and A. Sharieh, "Task scheduling algorithm in cloud computing based on modified round robin algorithm," Journal of Theoretical & Applied Information Technology, vol. 96(17), 2018.

[36] T. Saif, N. Javaid, M. Rahman, H. Butt, M. B. Kamal, and M. J. Ali, "Round Robin inspired history-based load balancing using cloud computing," In International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 496-508, October, 2018.

[37] K. Dubey, M. Kumar, and S. C. Sharma, "Modified HEFT Algorithm for Task Scheduling in Cloud Environment," Procedia Computer Science, vol. 125, pp. 725-732, 2018.

[38] V. Vignesh, K. S. Sendhil Kumar, and N. Jaisankar, "Resource management and scheduling in cloud environment," International journal of scientific and research publications, vol. 3(6), pp. 1-6, 2013.

[39] M. Kumar, and S. C. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," Computers & Electrical Engineering, vol. *69*, pp. 395-411, 2018.

[40] T. D. Braun, H. J. Siegel, N. Beck, L. L. Boloni, M. Maheswaran, A. I. Reuther, J.P. Robertson, M. D. Theys, B. Yao, D. hensgen and R. F. Freund, "A comparison of eleven static heuristic mapping a class of independent tasks onto heterogenous distributed computing systems," Journal of Parallel and Distributed Computing, vol. 61, no. 6, pp.810–837, 2001.

[41] X. He, X. Sun, and G. Von Laszewski, "QoS guided min-min heuristic for grid task scheduling," Journal of Computer Science and Technology, vol. 18(4), pp. 442-451, 2003.

[42] K. Etminani and M. Naghibzadeh, "A Min-min Max-min Selective Algorithm for Grid Task Scheduling," TheThird IEEE/IFIP International Conference on Internet, Uzbekistan, 2007.

[43] S. Parsa and R. Entezari-Maleki, "RASA: A new task scheduling algorithm in grid environment," World Applied Sciences Journal 7 (Special Issue of Computer & IT): pp. 152-160, 2009.

[44] S. Devipriya, and C. Ramesh, "Improved max-min heuristic model for task scheduling in cloud," International Conference on Green Computing, Communication and Conservation of Energy (ICGCE), pp. 883-888, December, 2013.

[45] H. Chen, F. Wang, N. Helian, and G. Akanmu, "User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing," National conference on Parallel computing technologies (PARCOMPTECH), pp. 1-8, February, 2013.

[46] S. K. Panda, and P. K. Jana, "Efficient task scheduling algorithms for heterogeneous multi-cloud environment," The Journal of Supercomputing, vol. 71(4), pp. 1505-1533, 2015.

[47] J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. 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(2), pp. 305-316, 2016.

[48] S. K. Panda, I. Gupta, and P. K. Jana, "Task scheduling algorithms for multi-cloud systems: allocation-aware approach," Information Systems Frontiers, pp. 1-19, 2017.

[49] J. K. Konjaang, F. H. Ayob, and A. Muhammed, "Cost Effective Expa-Max-Min Scientific Workflow Allocation and Load Balancing Strategy in Cloud Computing," 2018.

[50] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "Hybrid metaheuristics in combinatorial optimization: A survey," Applied Soft Computing, vol. 11(6), pp. 4135-4151, 2011.

[51] F. W. Glover, and G. A. Kochenberger, Handbook of metaheuristics. Springer Science & Business Media, vol. 57, 2006.

[52] C. R. Reeves, Modern heuristic techniques for combinatorial problems. Advanced topics in computer science, Mc Graw-Hill, vol. 15, 1995.

[53] C. Blum, and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," ACM computing surveys (CSUR), vol. 35(3), pp. 268-308, 2003.

[54] S. Voss, S. Martello, I. H. Osman, and C. Roucairol, Meta-Heuristics - Advances and Trends in Local Search Paradigms for Optimization, Kluwer Academic Publishers, 1999.

[55] M. Gendreau, G. Laporte, and F. Semet, "A Dynamic Model and Parallel Tabu Search Heuristic for Real-Time Ambulance Relocation," Parallel Computing vol. 27, pp. 1641–1653, 2001.

[56] M. Gendreau, F. Guertin, J. Y. Potvin, and E.D. Taillard, "Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching," Transportation Science, vol. 33, pp. 381–390, 1999.

[57] C. W. Tsai, and J. J. Rodrigues, "Metaheuristic scheduling for cloud: A survey," IEEE Systems Journal, vol. 8(1), pp. 279-291, 2014.

[58] C. W. Tsai, W. C. Huang, M. H. Chiang, M. C. Chiang, and C. S. Yang, "A hyper-heuristic scheduling algorithm for cloud," IEEE Transactions on Cloud Computing, vol. 2(2), pp. 236-250, 2014.

[59] Z. Beheshti, and S. M. H. Shamsuddin, "A review of population-based meta-heuristic algorithms," Int. J. Adv. Soft Comput. Appl, vol. 5(1), pp. 1-35, 2013.

[60] H. H. Hoos and T. St¨utzle, Stochastic Local Search: Foundations & Applications. Amsterdam, The Netherlands: Elsevier, 2004.

[61] S. Kirkpatrick, C. D. Gelatto, and M. P. Vecchi, "Optimization by simulated annealing", Science, vol. 220, pp. 671–680, 1983.

[62] F. Glover, and C. McMillan, "The general employee scheduling problem: an integration of MS and AI," Computers & Operations Research, vol. 13, no. 5, pp. 563-573, 1986.

[63] F. Glover, "Tabu Search - Part 1," ORSA Journal on Computing, vol. 1, no. 2, pp. 190–206, 1989.

[64] F. Glover, "Tabu Search - Part 2", ORSA Journal on Computing, vol. 2, no. 1, pp. 4–32, 1990.

[65] J.R. Koza, Genetic programming: on the programming of computers by means of natural selection. The MIT press, 1992.

[66] D.B. Fogel, Evolutionary computation: toward a new philosophy of machine intelligence. Wiley-IEEE Press, 2006.

[67] J. H. Holland, "Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence," Michigan, Ann Arbor, University of Michigan Press, 1975.

[68] G. Beni, and J. Wang, "Swarm intelligence in cellular robotic systems," In Robots and Biological Systems: Towards a New Bionics, Springer, Berlin, Heidelberg, pp. 703-712, 1993.

[69] Y. Zhang, P. Agarwal, V. Bhatnagar, S. Balochian, and J. Yan, "Swarm intelligence and its applications," The Scientific World Journal, 2013.

[70] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," Proceedings of IEEE International Conference on Neural Networks, IEEE Press, Piscataway, NJ, pp. 1942-1948, 1995.

[71] R. Eberhart, and J. Kennedy, "A new optimizer using particle swarm theory," In Micro Machine and Human Science, MHS'95., Proceedings of the Sixth International Symposium on IEEE, pp. 39-43, October, 1995.

[72] M. Dorigo, V. Maniezzo, and A. Colorni, "Ant System: Optimization by a colony of cooperating agents," IEEE Transactions on Systems, Man, and Cybernetics - Part B, vol. 26(1), pp. 29-41, 1996.

[73] M. Dorigo and T. Stützle, Ant Colony Optimization. MIT Press, Cambridge, MA, 2004.

[74] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Technical Report, TR06, 2005.

[75] D. Karaboga and B. Basturk, "Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems," Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, vol. 4529/2007 of LNCS: pp. 789-798, Springer, Berlin, 2007.

[76] H. Shah-Hosseini, "Problem solving by intelligent water drops", Proceedings of IEEE Congress on Evolutionary Computation, pp. 3226–3231, 2007.

[77] V. Kayvanfar, M. Zandieh, and E. Teymourian, "An intelligent water drop algorithm to identical parallel machine scheduling with controllable processing times: a just-in-time approach," Computational and Applied Mathematics, vol. 36(1), pp. 159-184, 2017.

[78] E. Rashedi, S. Nezamabadi, and S. Saryazdi, "GSA: a gravitational search algorithm," Information Sciences, vol. 179, no. 13, pp. 2232– 2248, 2009.

[79] X. S. Yang, "A new metaheuristic bat-inspired algorithm," In Nature inspired cooperative strategies for optimization (NICSO), pp. 65-74, Springer, Berlin, Heidelberg, 2010.

[80] X. S. Yang, "Firefly Algorithm, Stochastic Test Functions and Design Optimization," Int. J. Bio-Inspired Computation, vol. 2, no. 2, pp.78–84, 2010.

[81] I. Fister, Jr I. Fister, X. S. Yang, and J. Brest, "A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation, vol. 13, pp. 34-46, 2013.

[82] X. S. Yang, and S. Deb, "Cuckoo search via Lévy flights," In Nature & Biologically Inspired Computing, NaBIC, World Congress on, pp. 210-214, IEEE, December, 2009.

[83] R. B. Payne, and M. D. Sorensen, The cuckoos, vol. 15. Oxford University Press, 2005.

[84] A. H. Kashan, “League Championship Algorithm (LCA): An algorithm for global optimization inspired by sport championships,” Applied Soft Computing, vol. 16, pp. 171-200, 2014.

[85] B. Mondal, K. Dasgupta, and P. Dutta, “Load balancing in cloud computing using stochastic hill climbing-a soft computing approach,” Procedia Technology, vol4, pp. 783-789, 2012.

[86] Z. Fan, H. Shen, Y. Wu, and Y. Li, “Simulated-Annealing Load Balancing for Resource Allocation in Cloud Environments,” In PDCAT, pp. 1-6, December, 2013.

[87] D. Pandit, S. Chattopadhyay, M. Chattopadhyay, and N. Chaki, “Resource allocation in cloud using simulated annealing,” In Applications and Innovations in Mobile Computing (AIMoC), pp. 21-27, IEEE, February, 2014.

[88] B. Mondal, and A. Choudhury, “Simulated annealing (SA) based load balancing strategy for cloud computing,” International Journal of Computer Science and Information Technologies, vol. 6(4), pp. 3307-3312, 2015.

[89] Y. Ge, and G. Wei, “GA-based task scheduler for the cloud computing systems,” In Web Information Systems and Mining (WISM), 2010 International Conference on, vol. 2, pp. 181-186, IEEE, 2010.

[90] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, “A genetic algorithm (GA) based load balancing strategy for cloud computing,” Procedia Technology, vol. 10, pp. 340-347, 2013.

[91] T. Wang, Z. Liu, Y. Chen, Y. Xu, and X. Dai, “Load balancing task scheduling based on genetic algorithm in cloud computing,” In Dependable, Autonomic and Secure Computing (DASC), 2014 IEEE 12th International Conference on, pp. 146-152, IEEE, August, 2014.

[92] M. Vanitha, and P. Marikkannu, “Effective resource utilization in cloud environment through a dynamic well-organized load balancing algorithm for virtual machines,” Computers & Electrical Engineering, vol. 57, pp. 199-208, 2017.

[93] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, “A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments,” In Advanced information networking and applications (AINA), 2010 24th IEEE international conference on, pp. 400-407, IEEE, April, 2010.

[94] Z. Liu, and X. Wang, “A PSO-based algorithm for load balancing in virtual machines of cloud computing environment,” In International Conference in Swarm Intelligence, pp. 142-147, Springer, Berlin, Heidelberg, 2012.

[95] F. Ramezani, J. Lu, and F. K. Hussain, “Task-based system load balancing in cloud computing using particle swarm optimization,” International journal of parallel programming, vol. 42(5), pp. 739-754, 2014.

[96] A. I. Awad, N. A. El-Hefnawy, and H.M. Abdel_kader, “Enhanced particle swarm optimization for task scheduling in cloud computing environments,” Procedia Computer Science, vol. 65, pp. 920-929, 2015.

[97] F. Ebadifard, and S. M. Babamir, “A PSO-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment,” Concurrency and Computation: Practice and Experience, vol. 30(12), e4368, 2018.

[98] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, “Cloud task scheduling based on load balancing ant colony optimization,” In 2011 Sixth Annual ChinaGrid Conference, pp. 3-9, IEEE, August, 2011.

[99] R. Mishra, and A. Jaiswal, “Ant colony optimization: A solution of load balancing in cloud,” International Journal of Web & Semantic Technology, vol. 3(2), 33, 2012.

[100] K. Nishant, P. Sharma, V. Krishna, C. Gupta, K. P. Singh, and R. Rastogi, “Load balancing of nodes in cloud using ant colony optimization,” In 2012 14th International Conference on Modelling and Simulation, pp. 3-8, IEEE, March, 2012.

[101] S. Dam, G. Mandal, K. Dasgupta, and P. Dutta, “An ant colony based load balancing strategy in cloud computing,” In Advanced Computing, Networking and Informatics, vol. 2, pp. 403-413, Springer, Cham, 2014.

[102] M. Padmavathi, and S. M. Basha, “Dynamic and elasticity ACO load balancing algorithm for cloud computing,” In Intelligent Computing and Control Systems (ICICCS), 2017 International Conference on, pp. 77-81, IEEE, June, 2017.

[103] A. Gupta, and R. Garg, “Load balancing based task scheduling with ACO in cloud computing,” In Computer and Applications (ICCA), 2017 International Conference on, pp. 174-179, IEEE, September, 2017.

[104] P. V. Krishna, “Honey bee behavior inspired load balancing of tasks in cloud computing environments,” Applied Soft Computing, vol. 13(5), pp. 2292-2303, 2013.

[105] D. P. Mahato, and R. S. Singh, “Load balanced transaction scheduling using Honey Bee Optimization considering performability in on-demand computing system,” Concurrency and Computation: Practice and Experience, vol. 29(21), e4253, 2017.

[106] S. Elsherbiny, E. Eldaydamony, M. Alrahmawy, and A. E. Reyad, “An extended Intelligent Water Drops algorithm for workflow scheduling in cloud computing environment,” Egyptian informatics journal, vol. 19(1), pp. 33-55, 2018.

[107] L. Tang, Z. Li, P. Ren, J. Pan, Z. Lu, J. Su, and Z. Meng, “Online and offline based load balance algorithm in cloud computing,” Knowledge-Based Systems, vol. 138, pp. 91-104, 2017.

[108] S. M. Abdulhamid, M. S. A. Latiff, and I. Idris, “Tasks scheduling technique using league championship algorithm for makespan minimization in IAAS cloud,” arXiv preprint arXiv:1510.03173, 2015.

[109] N. Susila, S. Chandramathi, and R. Kishore, “A fuzzy-based firefly algorithm for dynamic load balancing in cloud computing environment,” Journal of Emerging Technologies in Web Intelligence, vol. 6(4), pp. 435-440, 2014.

[110] A. P. Florence, and V. Shanthi, “A load balancing model using firefly algorithm in cloud computing,” Journal of Computer Science, vol. 10(7), 1156, 2014.

[111] M. Yakhchi, S. M. Ghafari, S. Yakhchi, M. Fazeli, and A. Patooghi, “Proposing a load balancing method based on Cuckoo Optimization Algorithm for energy management in cloud computing infrastructures,” In Modeling, Simulation, and Applied Optimization (ICMSAO), 2015 6th International Conference on, pp. 1-5, IEEE, May, 2015.

[112] D. Chaudhary, and B. Kumar, “Cloudy GSA for load scheduling in cloud computing,” Applied Soft Computing, vol. 71, pp. 861-871, 2018.

[113] M. Abdullahi, and M. A. Ngadi, “Symbiotic Organism Search optimization based task scheduling in cloud computing environment,” Future Generation Computer Systems, vol. 56, pp. 640-650, 2016.

[114] C. Blum, J. Puchinger, G. R. Raidl, and A. Roli, "A brief survey on hybrid metaheuristics," Proceedings of BIOMA, pp. 3-18, 2010.

[115] G. N. Gan, T. L. Huang, and S. Gao, "Genetic simulated annealing algorithm for task scheduling based on cloud computing environment," In Intelligent Computing and Integrated Systems (ICISS), 2010 International Conference on, pp. 60-63, IEEE, October, 2010.

[116] H. M. Alkhashai, and F. A. Omara, "An Enhanced Task Scheduling Algorithm on Cloud Computing Environment. International Journal of Grid and Distributed Computing, vol. 9(7), pp. 91-100, 2016.

[117] R. Raju, R. G. Babukarthik, D. Chandramohan, P. Dhavachelvan, and T. Vengattaraman, "Minimizing the makespan using Hybrid algorithm for cloud computing," In Advance Computing Conference (IACC), 2013 IEEE 3rd International, pp. 957-962, IEEE, February, 2013.

[118] K. M. Cho, P. W. Tsai, C. W. Tsai, and C. S. Yang, "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing," Neural Computing and Applications, vol. 26(6), pp. 1297-1309, 2015.

[119] S. Torabi, and F. Safi-Esfahani, "A dynamic task scheduling framework based on chicken swarm and improved raven roosting optimization methods in cloud computing," The Journal of Supercomputing, vol. 74(6), pp. 2581-2626, 2018.

[120] S. Sobhanayak, A. K. Turuk, and B. Sahoo, "Task scheduling for cloud computing using multi-objective hybrid bacteria foraging algorithm," Future Computing and Informatics Journal, pp. 210-230, 2018.

[121] T. S. Alnusairi, A. A. Shahin, and Y. Daadaa, "Binary PSOGSA for Load Balancing Task Scheduling in Cloud Environment," arXiv preprint arXiv:1806.00329, 2018.

[122] Z. Zhang, and X. Zhang, "A load balancing mechanism based on ant colony and complex network theory in open cloud computing federation," 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Vol. 2, pp. 240-243, IEEE, May, 2010.

[123] S. Javanmardi, M. Shojafar, D. Amendola, N. Cordeschi, H. Liu, and A. Abraham, "Hybrid job scheduling algorithm for cloud computing environment," In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications IBICA, pp. 43-52, Springer, Cham, 2014.

[124] M. Shojafar, S. Javanmardi, S. Abolfazli, and N. Cordeschi, "FUGE: A joint meta-heuristic approach to cloud job scheduling algorithm using fuzzy theory and a genetic method," Cluster Computing, vol. 18(2), pp. 829-844, 2015.

[125] E. N. Alkhanak, and S. P. Lee, "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing," Future Generation Computer Systems, pp. 480-506, 2018.

Kaushik Mishra, currently pursuing his PhD in Computer Science and Engineering (CSE) at Veer Surendra Sai University of Technology, Burla, Sambalpur, India. He has completed his BTech and MTech in CSE. His research area includes Cloud Computing and IOT.

Dr. Santosh Kumar Majhi, currently working as an Assistant Professor in department of Computer Science and Engineering at Veer Surendra Sai University of Technology, Burla, Sambalpur, India. He has completed his BTech and MTech in CSE and PhD in Computer and Information Systems. His research area includes cloud computing, security analytics, data mining, soft computing and AI.