

CURRENT PERSPECTIVE IN TASK SCHEDULING TECHNIQUES IN CLOUD COMPUTING: A REVIEW

Jyoti Thaman, Manpreet Singh

Department of Computer Science & Engineering M. M. University, Ambala,
Haryana, India

ABSTRACT

Cloud computing is a development of parallel, distributed and grid computing which provides computing potential as a service to clients rather than a product. Clients can access software resources, valuable information and hardware devices as a subscribed and monitored service over a network through cloud computing. Due to large number of requests for access to resources and service level agreements between cloud service providers and clients, few burning issues in cloud environment like QoS, Power, Privacy and Security, VM Migration, Resource Allocation and Scheduling need attention of research community. Resource allocation among multiple clients has to be ensured as per service level agreements. Several techniques have been invented and tested by research community for generation of optimal schedules in cloud computing. A few promising approaches like Metaheuristics, Greedy, Heuristic technique and Genetic are applied for task scheduling in several parallel and distributed systems. This paper presents a review on scheduling proposals in cloud environment.

KEYWORDS

Metaheuristics, Heuristics, Greedy, Genetic Algorithms, Review, Task Scheduling.

1. INTRODUCTION

The cloud computing model is composed of three service models and four deployment models as shown in Figure 1. Clouds offer services that can be grouped into three categories: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). Deployment Models are classified as Public Clouds, Private Clouds, Community Clouds and Hybrid Clouds. This huge conglomeration of services and sources, shared among clients on subscription basis needs a serious attention in terms of tasks scheduling, resource allocation and resource sharing. Another important concerns for cloud service providers are energy efficiency and bandwidth management. If seen by outsider's view, a cloud environment processes the tasks submitted by clients. Any simultaneous access to resources needs to be addressed with objectives of improved resource utilization, reduced energy expenses and reduced makespan. Several proposals in recent times addressed scheduling issues in cloud environments.

In literature several promising and successful techniques and approaches have evolved as master tool for guaranteed optimal scheduling of tasks. Scheduling proposals in recent times can be classified as Metaheuristics, Heuristics, Greedy and Genetic Approach based solutions. Metaheuristics approach like Particle Swarm Optimization (PSO), Ant Colony Optimization

(ACO), BAT echolocation and Artificial Bee Colony (ABC) hasfound their places in scheduling of tasks in cloud environments. Several approaches like Simulated Annealing, Hill climbing etc. are classified as heuristic techniques. Heuristics techniques may not guarantee optimal solution but may provide incidental solution of some problems. Another class of algorithms used genetic operators for generation and searching local and global optimal solution over solution space. This category of algorithms may be classified as Genetic Algorithms (GAs). Greedy based solution to scheduling problem in cloud environment is another valuable approach. This paper has presents a review of many proposals in recent times and presents a novel taxonomy for classification of algorithms into Goal Oriented Task Scheduling or Constraints Oriented Task Scheduling.

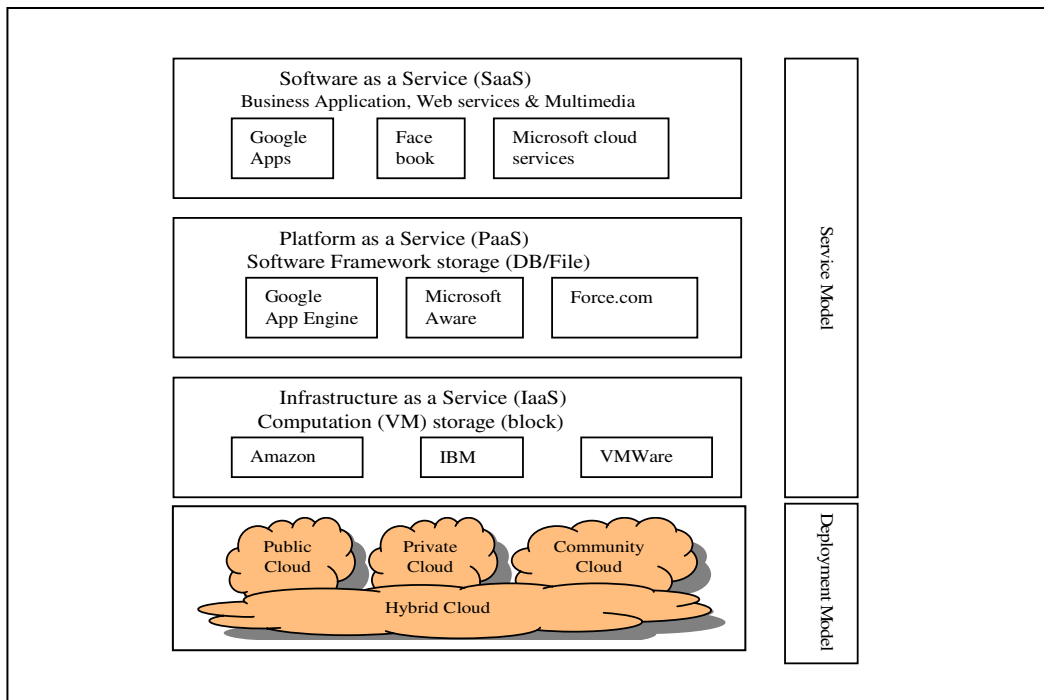


Figure 1 Cloud Platform

This paper is organized as follows. Section 2 presents a new taxonomy for classification of scheduling approaches. Section 3 presents a detailed review of scheduling schemes. Finally, Section 4 concludes the article.

2. TASK SCHEDULING TAXONOMY

Task scheduling is prime concern when a group of clients with specified Service Level Agreement (SLA) submits their jobs for processing to cloud over network. Tasks are submitted with certain constraints and deadline requirements. Specified constraints must be respected as per SLA. Any breach in SLA costs cloud service providers in terms of both economically and trustworthiness wise. To maintain high trustworthiness and reliability in service, Tasks must be scheduled with either goal or constraints in mind. When client doesn't specify any constraint, cloud service provider must strive to reduce the energy expenses and improve upon resource utilization. Such scheduling may be considered under Goal Oriented Task Scheduling (GOTS).

GOTS allows the cloud service providers to optimize the resources, by maximizing the resource usage, minimizing the delays and maximizing the bandwidth usage etc. GOTS allows cloud service providers to optimize their services and ensure maximum Return on Investment (ROI).

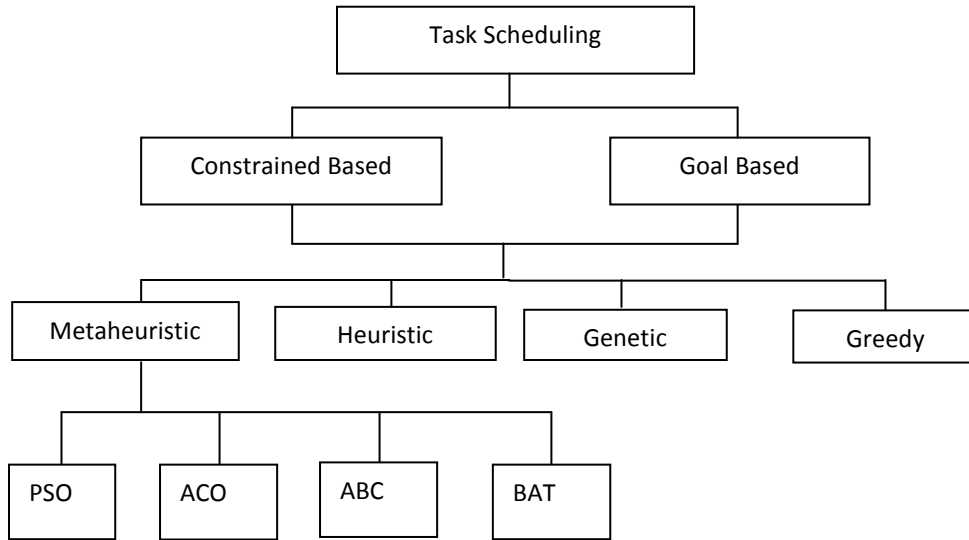


Figure 2 Task Scheduling Taxonomy

In case client specifies certain constraints like cost, deadlines and delays etc. cloud service providers must schedule tasks of clients with constraints in mind. This type of task scheduling is called Constraint Oriented Task Scheduling (COTS). COTS give a biased chance to cloud service providers to negotiate SLA with users on their own terms. Cloud service providers may strike a better deal in this scenario. The clients who submit their tasks with constraints are prioritized over other clients and treated as priority costumers. In brief, task scheduling schemes may be classified either as GOTS or COTS. Figure 2 presents a classification of task scheduling and present taxonomy for classification of task scheduling schemes in cloud environment.

Several classical and recent promising scheduling approaches are classified either as COTS or GOTS or both. Metaheuristics based solutions, heuristic based solutions, genetic algorithm based solutions and greedy approach based solutions are generally used to implement scheduling in cloud and grid computing environments. Depending upon the goals achieved or constraints honoured, this paper reclassified various recent solutions either as COTS or GOTS. This paper has reviewed various schemes under Metaheuristic, heuristic and genetic algorithm based approaches.

3. METAHEURISTIC BASED SCHEDULING

Metaheuristic based solutions are proposed using several distinct forms of metaheuristics. Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) and BAT's echolocation behavior (BAT) are most promising metaheuristic approaches used in task scheduling.

3.1 Particle Swarm Optimization (PSO)

PSO has been found to be robust and is successfully applied in solving nonlinear, non-differentiable multi-modal problems quickly. It is still in its infancy. Many research works have mentioned application of PSO in task scheduling. PSO is most successful metaheuristic for generations of optimal scheduling solutions. PSO scans over solution space during each iteration and accumulates global best and local best solutions. This section presents review of recent proposals which considered PSO in the field of task scheduling in cloud environment.

Originally PSO was proposed in [1] where PSO was proposed as an optimization tool. Two types of PSO namely, Discrete PSO and Continuous PSO versions were proposed. With several passes over the search space and updating local best and global best solutions during each pass, PSO performed much faster than ACO or GA. In [2] authors introduced the concept of inertia weight into the original PSO. With introduction of inertia weight PSO could converge even faster. Initially inertia weight was proposed to lie in the range [0.9, 1.2], which can improve performance of PSO. Different values of inertia allowed better control over solution search space. Higher values of inertia weight will result in overshooting the and lower values will trap search in definite area in search space. A Cost Aware Modified PSO (CA-PSO) was proposed in [3]. In [4] authors exploit PSO for optimizing overall tasks completion cost in a workflow and respecting the given deadline constraints. The proposed metaheuristic approach based on PSO succeeds whereas IC-PCP fails to meet application's deadline. In comparison IC-PCP failed to meet deadline constraints as IC-PCP ignored VM boot time. Results prove that PSO performs better than current state-of-the-art algorithms. Proposal considered deadline constraint. Proposal generates constraint makespan and performs cost evaluation for various workflows like Montage, Ligo etc. When compared to SCS, proposed algorithm is capable of generating better schedules and achieved cost optimization. In [5] authors proposed mathematical model using a Load Balancing Mutation (balancing) Particle Swarm Optimization (LBMP SO) and considered reliability and availability as the objective parameters of proposals. LBMP SO used an algorithm to generate schedule and allocation for cloud computing environment. Algorithm considered available resources for generation of schedule and allocation patterns. Basic PSO suffers from free VMs, allocation of more than one task to same VM, allocation of same tasks to multiple VMs and premature convergence. LBMP SO takes into account execution time, transmission time, make span, round trip time, transmission cost and load balancing between tasks and achieved reliability in task scheduling. Idea of LBMP SO is to reschedule failure tasks to available VM. LBMP SO performance was compared with standard PSO, random algorithm and Longest Cloudlet to Fastest Processor (LCFP) algorithm to show that LBMP SO can save in make span, execution time, round trip time, transmission cost. In [6], authors proposed a model for resource-task mapping which could reduce execution cost and also designed a PSO based heuristic to allocate tasks to resources. Both computation cost and data transmission cost are calculated by using the workflow application. Authors compared results of heuristic against "Best Resource Selection" (BRS) heuristic and found that PSO based task scheduling could result into three times cost savings. In [7], authors compared three popular heuristic approaches namely PSO, GAs and MP SO for efficient task scheduling in cloud environment. MP SO algorithm improved makespan characteristics when compared with PSO and GA. In [8], authors proposed a hybrid of particle swarm optimization and simulated annealing. CloudSim toolkit [add reference] was used to implement and analyze the proposal. Population based metaheuristic (PBM) algorithm maximizes resource utilization and minimizes makespan and demonstrating improvements upwards of 53%. Hybrid meta-heuristic method proved to enhance the performance in CloudSim toolkit. In terms of resources utilization and makespan Hybrid PSO implementation in [9] tried to balance the load

across the system and minimize the makespan. In [10], authors presented multi-objective PSO based optimization algorithm for dynamic environment of clouds and optimize energy and processing time. Proposed algorithm provides an optimal balance results for multiple objectives. Experimental results illustrated that proposed methods (TSPSO) out-performed BRS and RSA. In CA-PSO authors proposed Modified PSO by augmenting PSO with Cost-Aware Heuristic. CA-PSO generates tasks schedules using a cost aware fitness function to quantify the cost of resource usage along with fitness function for time cost to minimize processing time. Authors preferred to optimize usage cost of resources if possible. In [11], a task-level scheduling algorithm Chaotic Particle Swarm Optimization (CPSO) is based on chaotic sequence and inertia weight factor. Chaotic sequence with high predictability improves the variety of solutions and its reliability assures a good global convergence. It can optimize the cost of whole scheduling and overcome the premature convergence of PSO algorithm to satisfy the market-oriented characteristic of cloud workflow. The performance of CPSO is efficient and the cost is the lowest as compared with other representative counterparts.

3.2 Ant Colony Optimization (ACO)

ACO is available as metaheuristic in Computer Science and Operation Search problems. This is a probabilistic search method and can be applied to any problem which can be reduced to graphs and searching optimal paths through graphs. Since its inception in 1995, ACO has gone through several extensions and has seen wider acceptance in many of the engineering domains. ACO simulates the searching behaviour of ants through ant-colony in search of foods. Some of the popular extensions of ACO like, Elitist ant system, Max-min ant system (MMAS), Rank-based ant system, Continuous Orthogonal Ant Colony (COAC) and Recursive ant colony optimization. This section presents review of few proposals in ACO based scheduling solutions in cloud environment.

In [12] authors established superiority of Ant Colony Optimization (ACO) algorithms against heuristic approaches like FCFS and RR. Performance of ACO in task scheduling is appreciable when makespan of ACO based task scheduling is considered. In [13], Authors proposed ACO based system that increases the profit of cloud service provider by outsourcing task to external cloud when its resources are not enough. ACO outperforms the existing SLPSO (Self-adaptive learning PSO) and provides optimized scheduling. The ACO based system decreases response time (making it more responsive) and throughput as compared with existing SLPSO system. In [14], authors proposed ACO-LB (ACO based Load Balancing) to target the imbalance of task loads assigned to VMs during task scheduling. Heuristics based task scheduling schemes fails to consider the load imbalance. Another issue with scheduling in cloud is dynamic arrival rate of tasks. Solution proposed in this paper is able to adapt to dynamic environment of clouds. The simulation based analysis established performance abilities of ACO-LB in Load Balancing. In [15], MACOLB (definition) was introduced to balance system load and minimize the makespan of given tasks so that performance of overall system can be improved. MACOLB achieved better system load balance and reduced makespan as compared with ACO and MACO algorithms. Simulation's results shows that MACOLB algorithm outperforms MACO and ACO algorithms. In [16] authors proposed a Load Shared Ant Colony Optimization (LSACO) to balance the load across system. Proposal simulated across tasks and resources of varied types. Decision to share load is based on storage requirements of tasks. The proposal successfully reduced response time, improved system throughput and resource utilization. In [17], authors proposed a new constrained based workflow scheduling algorithm to maximize reliability of workflow. In this paper authors

proposed three novel workflow scheduling heuristics which are based on Ant Colony System. The aggregation of these heuristics minimized violations of reliability and constraints. Authors find feasible schedules for time and cost considerations in resource selection by two heuristics and other is used to enhance reliability. Simulation result on real world workflows shows that proposed method increases feasibility ratio and reliability for various pairs of deadline and budget. Authors also overcome instances of reduction in Grid profit by increasing feasibility ratio. In [18], Authors proposed Energy Aware Ant Colony Optimization that reduces the energy consumption for cloud datacenter and to find the hosts that consume least power based on their CPU utilization and schedule tasks. Performance is compared with existing techniques. Proposed method improves Make-span and conserves energy consumption in hosts up to 22%.

3.3 Artificial Bee Colony (ABC)

ABC was proposed by Karaboga in 2005. ABC was inspired by honey bees' behaviour. ABC simulates foraging behaviour of honey bees. Set of bees is called swarm and completes any tasks by using social behaviour and cooperation. Bees are specified to be of three types namely, employed bees, onlooker bees and scout bees. Employed bees search for food sources and onlooker bees perform selection of good sources by using the information conveyed by employer bees. Fitness function is specified to rank the quality of the food source. ABC uses fitness function to rank any solution over the solution space. Higher fitness values ranks higher quality. Scout bees are translated from a few employed bees, which abandon their food sources and search new ones. Scout bees are promoted as employed bees, which search newer food sources. The application of honey bees' behaviour through ABC algorithms has been tested in scheduling in many proposals and is reviewed in this section.

In [19], authors compared the performance of ABC optimization schemes with Evolutionary Algorithms, PSO, and Differential Equations etc. for assessment of ABC to multi-dimensional numeric problems. The behaviour of honey bees was imitated by using Artificial Intelligence. Simulation results show that ABC is equally applicable to multidimensional engineering problems. In [20] authors presented a new bee colony algorithm to schedule tasks on service providers in Cloud environments. The proposed algorithm has better results of simulation in terms of task execution time, waiting time and missed tasks. In [21] the Honey bee forage mechanism was proposed for load balancing in cloud to utilize its resources in cloud and was applied to optimize scheduling of Virtual Machine (VM) and reducing makespan, time a task must be held in a queue on VM. The proposed algorithm reduced makespan of processing time and improves general turnout of process and priority based balancing. In [22] authors presented an ABC Algorithm that computes total finishing time, mean finishing time, load balancing time after scheduling task. Authors considered time as main QoS factor. Algorithm produced best total task finishing time, mean task finishing time and load balancing time as compared with ACO. In [23], authors proposed Two-stage Artificial Bee Colony (TABC) algorithm for scheduling and re-scheduling with new job(s) inserting. The algorithm is compared with six simple heuristics and proposed hybrid heuristics. TABC adds crossover operator and critical path based local search for solving flexible job-shop scheduling problem (FJSP) problem. The proposed algorithm obtains better results in all 30 repeats than compared simple and combination eight heuristics in both initializing scheduling and re-scheduling phase. Result shows TABC algorithm has very good convergence and stability and to minimize makespan. In [24] proposed the hybrid algorithm TABC a combination of artificial bee colony (ABC) and TabuSearch (TS) algorithm. TS based strategy was embedded in TABC to solve the large-scale hybrid flow shop scheduling problem with limited buffers and was tested on sets of instances with large scales that are generated based

on realistic production. The proposed algorithm balanced exploitation and exploration abilities of the algorithms and TS-based local search is applied to the employed bees and onlookers with different functions. In [25] authors compared the ABC algorithm with GA to block a solution from allocating colossal tasks on a single machine and also to maintain balance of work allocation among machines. Author found that the ABC performed better than GA to minimization of maximal machine workload, represented by maximum runtime consumed by any machine and symbolized by overall run time allocated to all the machines. It also reduced make span of tasks as compared with Genetic.

3.4 BAT Optimization (BAT)

In [26] authors proposed a new metaheuristic approach and also included the good properties of other approaches. The behavior of this optimization is based on echolocation behavior of BATs. BAT uses a special type of SONOR called echo-locater to locate their hives, obstacles etc. BATs generate a sound which echoes back and is used by BATs to locate the obstacles. Each sound burst last 10-20 milliseconds but the waves have constant frequency between 25-150MHz. Micro bats can generate 10-20 such sound burst and can be increased upto 100-200 burst per second, when BATs chase any prey. Some bats have good eyesight and some have good smell senses. The overall effort is to combine all these and obtain an effect which gives efficient prey detection and a smooth navigation. Such echolocation function was associated with objective function to be optimized and a new optimization algorithm called Bat Algorithm (BA) was proposed. BA was composition of PSO and intensive local search controlled by loudness and pulse emission rate. The proposal was compared with Genetic Algorithms (GAs) and PSO. PSO performed better in terms of efficiency and accuracy than GAs, while BAT algorithm performed superior of all. BA added the best feature of PSO and new feature of bats echolocation. In [27] authors introduced chaos into BA to increase its global search mobility for robust global optimization. CBA selected different chaotic systems to replace the parameters in BA and used these different methods for potentially efficient alternatives to pseudo random sequences. The success rate of the CBAs shown in Statistical results proved that the improved algorithms can improve the reliability of the global optimality and quality of the results. In [28] presented a novel BCO based approach, namely JDS-BC, to schedule jobs to CNs and replicate datafiles in SNs. JDS- BC focuses on two independent matchmaking process and to minimized makespan and total transfer time concurrently in system.

Table 1 Metaheuristic Scheduling Scheme Parametric Analysis

S. No.	Constrained	Goal	Scheduling Factor/Parameter	Task Type	Tool	Achieved	Gaps patched/ Contribution
[2]		1. Global optimum. 2. Reasonable number of iterations	Based on varying inertia weight in range [0.9,1.2]	I	C Language	1. Improved PSO Performance 2. Inertia Introduced	1. PSO without inertia fails to find the global optimum within 4000 iterations
[3]		1. Minimize processing time 2. Maximize resource usage	1. Cost-Aware Based PSO 2. Scheduling Algorithm	I	1. MatLab 2. CloudSim	1. Task scheduling 2. Global optimal solution	1. Earlier PSO failed to consider the resource usage cost
[4]	1. Deadline	-	1. PSO_HOM (PSO based heuristic)	W	1. CloudSim	1. Minimize Execution Cost 2. Meeting	1. Performance Comparison with other

						Defined Deadline	Optimization Strategies and PSO
[5]	1. Reliability 2. Availability	-	1. Load Balancing 2. Mutation 3. PSO	I	1. CloudSim	1. Minimize cost 2. Minimize RTT 3. Minimize Execution Time 4. Minimize Transmission Time 5. Load Balancing 6. Improved Reliability 7. Resource Availability	1. Find the Data Transfer Cost between Data Centers so that VMs can be deployed in different regions. 1. Only Uniform VM Characteristic have been considered.
[6]	-	1. Computation Cost 2. Data Transmission Cost	1. PSO based Heuristic	W	1. Java based JSwarm package	1. Three times Cost Savings as compared to BRS	1. Integrate PSO based heuristic into Brain Imaging Analysis
[7]	-	1. Optimal schedule	1. Modified PSO	I	1. Eucalyptus Cloud	1. Reduced Makespan	1. Evaluated multiple System Performance Parameters
[8]	-	1. Minimize makespan 2. Increase Resource Utilization	1. Population-Based Metaheuristic (PBM) algorithm 2. Simulated Annealing	I	1. CloudSim	1. Improvement upwards of 53% in makespan 2. Resource Utilization	1. Implementation of a multi-objective solution
[9]	-	1. Load Balancing	1. Hybrid PSO	I	1. Cloudsim	1. Minimize Makespan 2. Improved Resource Utilization	1. Integrate HPSO inmap-reduce framework for Task Scheduling
[10]	-	1. Energy Efficiency	1. Nested Particle Swarm Optimization (TSP SO)	W	1. CloudSim	1. Optimization Energy Consumption 2. Overall Computation Time	1. Bandwidth usage Analysis 2. Load Balancing 3. Cost Etc.
[11]	-	1. To reduce the cost of Workflow system	1. Chaotic Particle Swarm Optimization (CPSO) algorithm	W	1. Workflow Sim	1. Improves the diversity of solutions and	1. Experiment with more

						good global convergence	than 4 VMs not be conducted
							2. QoS constraints are not considered
[12]	-	1. Minimize Makespan	1. Ant Colony Optimization (ACO)	I	1. CloudSim	1. Minimize Makespan	1.Task Precedence of Tasks 2. Load Balancing 3.Compared with other Metaheuristic s
[13]		1. Maximize Profit 2. QoS	1. Ant Colony Algorithm (ACO)	I	1. CloudSim	1.High Throughput 2.Improved Economy by 0.26-11.58%	1. Swarm Intelligence and ACO Hybridization
[14]	-	1. Load Balance of VMs	1. Load Balancing based Ant Colony Algorithm (ACO-LB)	W	CloudSim	1.Reduced Makespan 2.Increase Utilization 3.Balance Load	1.Different types of workflows were used to verify the reliability of the ACO-LB algorithm 2.Cost Problem 1.Cost 2.Task Types
[15]	-	1. Load Balancing	1. Modified Ant Colony Optimization for Load Balancing (MACOLB)	I	1. CloudSim	1.Minimize Makespan 2.Optimal Resource Allocation 3.Balance Load	1.Cost 2.Task Types
[16]	1. Memory Constrained	-	1. Load Shared Ant Colony Optimization (LSACO)	I	Java	1.Reduces Response Time 2.Increases Resource Utilization 3.Increased the no. of tasks scheduled	1. Considered ETC matrix as Parameter to get the Perfect Wait Time And Response Time
[17]	1. ReliabilityDriven Scheduling	-	1. Constrained Workflow Scheduling (CWS) algorithm	W	GridSim	1.Maximize the reliability of workflow Execution 2. Meet user defined Budget and Deadline 3. Reduces the Grid Profit	1. Variance in Load of Resources can change the Execution Time and Cost of activities

						Loss by increasing the Feasibility Ratio	
[18]	-	1. Energy Efficiency	1. Energy-AwareAnt Colony Algorithm (ACO)	I	CloudSim	1.Improves Energy Savings up to 22% 2.Improves Makespan	1. Implement IO Bound and Dynamic Workload 2. Energy Management in the Cloud
[20]		1. Tasks Scheduling	1. New Bee Colony Algorithm	W	Cloudsim	1.Reduced Task Execution Time 2.Reduced Waiting Time	1. Task Scheduling
[21]	-	1. Load Balancing	1. Honey Bee Galvanized Load Balancing Algorithm	I	CloudSim	1.Reduce Makespan 2. ReducedResponse Time	1. Considered alternative QoS factors of Tasks
[22]	-	1. Task Scheduling	1. ABC	I	CloudSim	1. Minimize Makespan 2. Mean Makespan 3. Load Balancing Time	1. Improved Datacenter Efficiency by adding more Parameters
[23]	-	1. Flexible Job-Shop scheduling Problem (FJSP)	1. Two-stage Artificial Bee Colony (TABCO)	I	C++	1. Minimize Makespan	1. Additional local Search Heuristics
[24]	-	1. Large-scale Hybrid Flow Shop scheduling problem with Limited Buffers (HFS-LB)	1. Hybrid Artificial Bee Colony (HABC)	I	C++	1. High Exploitation and Exploration capabilities in solving Large-scale HFS-LB	1. EnhancedBalance between Exploitation and Exploration 2. Reliability
[25]	-	1. Reduce Make-span	1. Artificial Bee Colony (ABC)	I	MATLAB	1.Minimize Makespan 2.Minimize Workload on any VM 3. Minimize Workload on all machine	1. This process can be analyzed with Parallel Algorithms
[26]	-	1. Improved Algorithmic Convergence	1. BAT Algorithm (BA)	I	MATLAB	1. Accuracy 2. Efficiency	
[27]	-	1. Increase	1. Chaos based	I	-	1. Improves	

	Global Search Mobility	BAT Algorithm (CBA)			Reliability	2. Enhance the quality of results	
[28]	-	1. Job Data Scheduling	1. Job Data Scheduling using Bee Colony (JDS-BC)	I	GridSim	1. Makespan 2.Total Data file Transfer Time	1. Energy Efficiency of Grid Resources 2. Cost of System Maintenance

***I= Independent, W = Workflows and B = Batch Jobs**

3. GENETIC ALGORITHM (GA) BASED SCHEDULING

GA is proved to be an effective method for searching over complex and difficult search space. GAs is collection algorithms which uses genetic operators like crossover, mutation and selection etc. GAs integrates other problem solving techniques with genetic operators. Combing the other approach with genetic operators generates Genetic Algorithms. GAs suffers from premature convergence of and may produce only local optimum values. This limitation of GA guides for hybridization of GAs with other techniques like metaheuristics and propose context specific hybrid GAs.

4.1 Current and Historic GAs Proposals:

In [29] authors proposed a genetic algorithms- (GAs-) based approach that effectively solved the problems of multi objective optimization. A genetic algorithms-based approach can be used to make decisions on how to assign tasks to cloud nodes and how to migrate task in pervasive clouds. Allocation strategies are used for optimized solutions and system can make use of the chosen allocation scheme to execute user tasks. The GA based approach provides the best solution for task migration in cloud environment. In [30] authors have proposed a GAbased scheduling heuristic for solving budget and deadline constrained optimization problems and scheduling workflow applications by either minimizing the monetary cost, execution time while meeting users' budget constraint and deadline constraints. Approach was compared with other heuristics on two workflows; balanced and unbalanced structures and results proved that GA performed better in handling a complex workflow structure. A Hierarchic Genetic Scheduler (HGS-Sched), a 2-level hybrid of Genetic Algorithm and Tabu Search (GA + TS) strategy for improving the efficiency of the single-population genetic-based schedulers in the dynamic grid environment was proposed in [31]. The HGS-Sched enabled a simultaneous search of solution space by many small dependent populations. In comparison with mono-population and hybrid genetic-based schedulers the proposed algorithm proved highly effective and resulted in fast

reductions in makespan and flowtime. In [32] authors proposed Multiple Priority Queues Genetic Algorithm (MPQGA) which incorporated a genetic algorithm (GA) approach which is suitable for scenario of Directed Acyclic Graph (DAG) scheduling designed with Crossover, Mutation and Fitness function. Proposal exploited the improvement of both Evolutionary-based and Heuristic-based algorithms while avoiding their drawbacks. Proposed MPQGA algorithm outperformed two non-evolutionary heuristics and a random search method in terms of schedule quality and covered a larger search space without incurring high computational cost than deterministic scheduling approaches. In [33] authors proposed a GA based load balancing strategy to balance the load and minimizing the makespan of given tasks provide an efficient utilization of resource in Cloud Computing infrastructure. In comparison with FCFS and RR and a local search algorithm Stochastic hill climbing, proposed GA algorithm outperform in terms of load balancing but also guarantees the QoS requirement of customer job. In [34] authors proposed Task scheduling algorithm using Improved GA. Authors compared output of improved GA with existing GA and found that improved GA reduced overall execution time of the tasks and improves utilization of resources.

Table 2 Genetic Algorithms Scheduling Schemes Parametric Analysis

S. No.	Constraint	Goal	Scheduling Factor/Parameter	Task Type	Tool	Achieved	Gaps Patched/Contribution
[29]	-	1. Optimized Tasks Allocation Scheme	1. GA based Approach	I	CloudSim	1. Fastest Execution Efficiency 2. Improve Task allocation and task migration efficiency	1. Dynamic Decision according to Cloud Environment 2. User's tasks Status 3. Context-awareness Framework
[30]	1. Deadline 2. Budget	-	1. GA based Approach	W	GridSim	1. Reduced execution Cost 2. Reduced Time	1. Enhanced scheduling algorithm by using different service models 2. Dynamic Data-driven workflow models
[31]	-	1. Improving Effectiveness of Single-Population GA based schedulers	1. Hierarchic Genetic Scheduler (HGS-Sched)	I and B	1. Sim-G- Batch grid	1. Minimized makespan 2. Flowtime	1. Improvement of effectiveness of Pareto front under Multi Objective Grid Scheduling
[32]	-	1. Task Scheduling on Heterogeneous Computing Systems using	1. Multiple Priority Queues GA (MPQGA)	W	C#	1. Reduced Makespan 2. Improve Speedup 3. Efficiency	1. Considered More Task graphs 2. More Processors 3. Variable Degrees of Heterogeneity among Processors and Subtasks

[33]	-	Multiple Priority Queue	1. Load Balancing	1. GA	I	Cloud Analyst	1. Load Balancing 2. Minimizing Make-span	1. Variation of the Crossover and Selection
[34]	-	1. Task Scheduling	1. GA	Improved GA	I	CloudSim	1. Reduced Makespan	1. Reduced Makespan

*I= Independent, W = Workflows and B = Batch Jobs

5. GREEDY APPROACH BASED SCHEDULING SCHEMES

Greedy approach is powerful and time tested approach. Several problem domains like shortest paths, spanning trees, activity selection etc. are popularly solved by using greedy based algorithms. Greedy algorithms are applied to problems with certain features. First requirements problem substructure, i.e., problem can be divided into sub-problems. Another requirement is that there exists a greedy choice property. Greedy choice property lets next local best step. In general, one of the sub-problems turns empty and other one non-empty. All in all there is only one sub-problem plus last local best step. Several proposals used greedy approach to solve scheduling problems in cloud environment.

In [35] authors proposed Greedy Method based Algorithm for cloud environment for improving makespan performance and user satisfaction. In [36] authors presented a greedy based approach to optimize scheduling which addressed the major challenges in cloud scheduling. Two level of job scheduling and rescheduling was performed by considering parameters like deadline, cost, profit, response time, waiting time and turnaround time etc. Greedy technique was used to select resources on the basis of task constraints. Re-Scheduling is based on cost/profit parameters. In comparison with existing algorithms like FCFS, Round Robin (RR) and Priority algorithms, the proposed algorithm improved the efficiency. Greedy technique is used to select resources on the basis of task constraints and improved the efficiency and resource utilization. In [37] authors used greedy approach to select the appropriate resource and minimize their execution cost in comparison of sequential approach and also reduced execution time of tasks. In [38] authors presented greedy based task scheduling to improve turnaround time performance and reduce cost incurred for user submitted tasks in particular time slot. Proposed algorithm selected the resource on the basis of its cost and turnaround time. Priority formula is used for task selection. Proposed algorithm gives better results over sequential scheduling. In [39] proposed Min-Max algorithm to verify its validity with Min-Min and Max-Min. It puts together small, big tasks and makes good use of time of greedy strategy in order to solve problem of load imbalance. The Min-Max results proved that saved up to 9% overall execution time compared with Min-Min and improved the utilization rate of the entire system compared with Max-Min and the total completion time and average response time reduced by 7% and 9% respectively. In [40] proposed task scheduling algorithm which respects the constraints specified for the said task. Authors adopted greedy method like activity selection using greedy approach. The proposed algorithm reduced turnaround time and cost characteristics of each task submitted.

Table 3 Greedy Algorithms based Scheduling Schemes Parametric Analysis

Sr. No.	Constraint	Goal	Scheduling Factor/Parameter	Task Type	Tool	Achieved	Gaps /Contribution
[35]	-	1. Job Scheduling	1. Greedy-Based job scheduling	I	CloudSim	1.Reduced Execution Time 2.Improved QoS	1. Improvement of the proposed algorithm in order to gain more fairness
[36]	1. Deadline 2. Cost	-	1. Greedy Algorithm	I	CloudSim	1. Improved Efficiency 2. Resource Utilization	1. More parameters for evaluation
[37]	-	1. Task Scheduling 2. Resource Allocation	1. Greedy algorithm	W	Eucalyptus	1. Reduced Execution Time 2. Cost	1. Cost based scheduling policy can be implemented
[38]	-	1. Task Scheduling	1. Greedy Algorithm	W	CloudSim	1. Reduced Task Completion Time 2. Execution Cost	1. Add location parameter with the existing Deadline and Cost
[39]	-	1. Task Scheduling	1. Min-Max	I	CloudSim	1. Improves Utilization 2. Task Completion Time 3. Average Response Time	
[40]	-	1. Task Scheduling	1. Dynamic Optimization Algorithm based on Greedy Approach	I	CloudSim	1. Reduced Turnaround Time 2. Cost	1. Type of task 2. Task Length could be taken into account

*I= Independent, W = Workflows and B = Batch Jobs

6. HEURISTIC BASED SCHEDULING

Heuristic Techniques is some practical and non-guaranteed method of problem solving in the field of science, technology etc. Heuristic technique may not guarantee the optimal solution but may prove helpful for immediate solution of the problem at hand. Heuristics are generally derived from rule-of-thumb, common sense, intuitive guess or educated judgements. Several proposals have used Heuristic based Approach for scheduling problems in cloud environments.

Authors in [41] proposed three different heuristics to assess importance of task scheduling in MapReduce. The parameter of importance is makespan. The scheduling problem was modeled as two-stage hybrid flowshop scheduling problem with schedule-based setup times. Jobs and tasks are considered two distinct entities. Two of the heuristics proposed in the paper are classified as job sequencing heuristics and one as task sequencing heuristic. Authors incorporated data locality of tasks in scheduling of tasks. Authors also proposed a lower bound of the makespan. The performance of heuristics depends upon number of jobs and cluster setups. Simulation based study compared performance of three heuristics and data dependent modeling is proved effective. In [42] authors proposed a heuristic based on cost as QoS parameter for optimization. The

experiment was simulated using cloudsims3.0 toolkit and shown improved cost effective performance. QoS driven algorithm considered execution time, load and latency as important parameters. The performance was compared against FCFS. In [43] authors proposed a scheduling heuristic to accommodate task priority, task length and user satisfaction. The historic solutions like min-min don't consider the task priority and thus ignored the satisfaction of user submitting the tasks to cloud environment. Authors proposed an approach to assign credits to tasks on the basis of task lengths and task priority. This has helped the cloud service providers to deal with tasks with same priority. Task credits obtained in the proposal are unique in the set of the task submitted for perusal. The performance of the credit based task scheduling was compared with that of scheduling algorithm with priority only and length only credits. In [44] authors presented a Maximum Effective Reduction (MER) algorithm which trades minimal increase in makespan against maximal resource usage reduction on the schedules generated by any given scheduling algorithm. Novelty of the approach is near-optimal trade off point determination. The tradeoff point consideration leads to reduced resource provisioning, improving resource utilization and reduced energy expenses. Another broad applicability of MER lies in the fact that it can be applied to any scheduling environment dealing with scientific workflows. MER works by consolidating tasks from over provisioned under- utilized resource to other resources and identify to reduce resource provisioning. The performance of MER was compared with Critical Path First (CPF), Critical-Path-On-Processor (CPOP) and Dynamic Critical Path (DCP). The exhaustive simulation reports about 52% of resource usage reduction against 10% increment in makespan. In [45] the authors reviewed several taxonomies for classification of challenges in Work Flow Scheduling schemes. The taxonomies are specified on the basis of QoS system functionality and system architecture. Authors also presented schematic on selecting the proposals suitable for inclusion in the review. Authors also identified future directions on the basis of schematic pursued. In [46] authors proposed a heuristic called Cost-Greedy Price-Adjusting (CGPA) for achievement of multiple objectives namely; System throughput, Total Cost and Profit Fairness. The heuristic benefits both grid users and providers. CGPA lowers the Fairness Deviation of Profits which is not generally taken care of by other popular algorithms. The algorithms also specified by the authors and simulation based performance of CGPA has been compared with four other popular heuristics. A fair amount of experimentation was carried out and leads to achievement of all the objectives. In [47] proposed a Constrained Workflow Scheduling (CWS) algorithm that is based on Ant Colony Optimization (ACO) to maximize the workflow execution reliability while respecting the user-defined deadline and budget. The author proposed three novel heuristics which are adaptively selected by ants. Two of them are employed to find feasible schedules and the other is used to enhance the reliability. The capability of CWS to find feasible schedules and compared with greedy approach toward optimization of time, reliability and cost of activities. The proposed algorithm has improved the feasibility ratio for various constraints, reliability and reduced the Grid profit loss. In [48] authors proposed community aware scheduling algorithm (CASA) that is one of the best dynamic scheduling approach are designed to efficiently distribute jobs amongst participating nodes. In cloud environment the CASA is comprised of two phase scheduling that is set of heuristic sub-algorithms to achieve optimized scheduling performance. The development made by an algorithm in this field of cloud has made a great contribution of task scheduling. In a decentralized scheduling manner without requiring detailed real-time processing information the use of CASA can lead to a 30%–61% better average job slowdown and a 68%–86% shorter average job waiting time compared with two reference conventional scheduling scenarios. The proposed community-aware scheduling algorithm (CASA) has improved performance when it works with faster response time information system, as compared with a slow reaction speed information system with integrated global knowledge. In [49] authors proposed a new Pareto-based multi-objective workflow scheduling

algorithm as an extension to an existing state-of-the-art heuristic capable of computing workflow schedules representing tradeoffs between energy consumption and makespan. This approach captures the real behavior of energy consumption that is based on empirical models in heterogeneous parallel systems. The proposed algorithm compared with a classical mono-objective scheduling heuristic and state-of-the-art multi-objective optimization algorithm and finding it computes better or similar results in different scenarios. The MOHEFT compared with HEFT and green HEFT, an ad-hoc greedy algorithm under different experimental configurations and observes it reduced the energy consumption by up to 34.5% with a slight increase of 2% in the makespan. In [50] proposed a Cloud-aware data intensive workflow scheduling system considered the QoS constraints of each workflow and improves the system performance by using Cloud resources by realistic scientific workflows. The proposed algorithm partitioned a workflow into sub-workflows to minimize data dependencies among the sub-workflows to distribute on volunteer resources. The proposed provisioning algorithm (VOLNT-SPOT) compared the performance with the system that contains just volunteer resources (VOLNT) as it improves the system performance by increasing the percentage of workflows that meet the deadline with a factor of 75% in average with respect to the execution of workflows on the volunteer resources. Two Policies are considered in deploying public Cloud resources significantly to increase the percentage of workflow in which first policy considers 23% saving in cost average than the second policy while the second strategy just considers the relaxed deadline by a larger margin. In [51] proposed a hybrid approach for tasks scheduling in Heterogeneous Distributed Computing (HySARC2) and used hierarchical clustering of the available resources into groups. Authors considered different DAG scheduling strategies for independent task. Proposed algorithm is suitable for Heterogeneous Distributed Computing, especially for modern High-Performance Computing (HPC) systems in which IO and computational intensive requirements are modeled. The HySARC2 algorithm more efficient and exact structure of resources because it analyzed the available resources and groups them into clusters (resource aware algorithm) and prerequisite different groups of similar tasks to different clusters of resources and then scheduled the tasks in each cluster of resources. It improves workload on the resources in cloud environment with respect to load-balancing, cost savings, dependency assurance for workflows.

Table 4 Heuristic based Scheduling Schemes Parametric Analysis

Sr. No.	Constraint	Goal	Scheduling Factor/Parameter	Task Type	Tool	Achieved	Gaps/Contribution
[41]	-	1. Job Scheduling 2. Task Scheduling	1. Job Sequencing Heuristic 2. Task Sequencing Heuristic	I	MapReduce	1. Lower bound on Makespan 2. Data Dependency Established 3. Cluster Set updependency Established	1. Data dependency incorporated 2. Lower bound on Makespan
[42]	-	1. Cost 2. Execution Time	1. Task Scheduling	I	CloudSim	1. Cost based Analysis	1. More parameters for evaluation can be added
[43]	-	1. Task Scheduling 2. Makespan	1. Credits based Scheduling	I	CloudSim	1. Improved Makespan Characteristics 2. Credit based on	1. Varying the VM characteristic could

[44]	-	1. Schedule Optimization 2. Reduced resource Provisioning 3. Improved Resource Utilization	1. Maximum Effective Reduction Heuristics	W	CloudSim	Length and priority of tasks 1. Reduced Resource Provisioning 2. Improved Utilization 3. Task Consolidation	have been considered 1. Optimized schedules generated by other heuristics
[45]	-	1. Cost Aware heuristic taxonomies 2. Future Directions	1. Taxonomy for cost aware Workflow Scheduling	W	-	1. Taxonomy considering QoS 2. System Functionality 3. System Architecture	-
[46]	-	1. Profit Fairness 2. Total Cost 3. System Throughput	1. Cost Greedy Price Adjusting (CGPA) Heuristic	W	-	1. Lower Fair Profit Deviation 2. Improved Throughput	1. Fair Allocation in terms of resource and cost
[47]	1. Deadline 2. Budget	-	1. Constrained Workflow Scheduling (CWS) algorithm	W	Grid Sim	1. Increases feasibility ratio 2. Reliability for various pairs of deadline and budget	1. Other kinds of constrained workflow scheduling can be investigated. 2. The variation in load of resources can change the execution time and cost of activities.
[48]		1. Decentralized Dynamic Scheduling	1. Community-Aware Scheduling Algorithm (CASA)	I	MaGate Sim	1. Faster Response Time 2. Improvement in both the average job slowdown and average job waiting time 3. Improves Resource Utilization	1. Implementation of CASA within the context of current developments in cloud computing
[49]	-	1. Makespan 2. Energy Efficiency	1. Multi-Objective Heterogeneous Earliest Finish time algorithm (MOHEFT)	W	POV-Ray	1. Reduced Energy 2. Makespan	1. Optimize Economic Cost 2. Extend MOHEFT

							to consider multi-tenant commercial cloud systems.
[50]	1. QoS2. Deadline	-	1. Cloud-Aware Data Intensive Workflow Scheduling	W	Cycloid Grid	1. Deadline 2. Cost	1. Volunteer resources 2. Decreases the number of Amazon spot instances
[51]	-	-	1. Hybrid approach for tasks Scheduling in Heterogeneous Distributed Computing (HySARC2)	W	Cloud Sim	-	1. Dynamic behavior of the resources and algorithms 2. Allow Tasks Preemption

7. CONCLUSION

This paper presents a review of Task and Job Scheduling schemes. A novel taxonomy is proposed in the paper. Schemes falling under Goal Oriented Task Scheduling (GOTS) schemes give service providers a fair chance to apply approach and schedule the tasks and resources that can generate maximum possible economic gains, while using least resource provisioning. Using low resource provision allows providers to use their resources at possible fullest and trading makespan with marginal increase only. Certainly GOTS is provider friendly policy. At the same time Constraint Oriented Task Scheduling (COTS) allows the user to specify their priority and importance. To ensure that desired level of service requests are met by service providers, SLAs are used. Users can specify their QoS parameters and other requirements. This paradigm gives providers an opportunity to negotiate the agreements at their own costs. COTS allow users to enjoy the flexibility and providers to earn higher. Table 1, 2, 3 and 4 gives a tabular glimpse of recent trends in scheduling schemes for cloud environment. With extensive review of recent proposals in scheduling schemes, we concludes that a lot more dimensions are yet to be explored in scheduling, like data dependency, two stage optimization of already generated schedules, batch jobs, Workflow technologies etc.

REFERENCES

- [1] <http://www.swarmintelligence.org/tutorials.php>, accessed on May 14, 2015.
- [2] Y. Shi and R. Eberhart, "A modified particle swarm optimizer", Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on. IEEE, (1998).
- [3] Zhao, G. (2014). 'Cost-Aware Scheduling Algorithm Based on PSO in Cloud Computing Environment', International Journal of Grid and Distributed Computing. 7, 1, .33-42

- [4] Rodriguez, M. A., &Buyya, R. (2014). Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *Cloud Computing, IEEE Transactions on*, 2(2), 222-235.
- [5] Awad, A. I., El-Hefnawy, N. A., &Abdel_kader, H. M. (2015). Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments. *Procedia Computer Science*, 65, 920-929.
- [6] Pandey, S., Wu, L., Guru, S. M., &Buyya, R. (2010, April). 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.
- [7] Abdi, S., Motamedi, S. A., &Sharifian, S. (2014). Task scheduling using Modified PSO Algorithm in cloud computing environment. In *International Conference on Machine Learning, Electrical and Mechanical Engineering* (pp. 8-9).
- [8] Hussein, S., Robert, C., Alam, M. (2014). 'Cloudlet Scheduling with Population Based Meta Heuristics'. *Journal of Computer Science and Management Research*. 1, 3, 348-351
- [9] Igomathi, B., Krishnasamy, K. (2013). 'Task scheduling algorithm based on hybrid particle swarm optimization in cloud computing environment'. *Journal of Theoretical and Applied Information Technology*. 55, 1, 33-38.
- [10] Jena R. K. (2015). Multi objective Task Scheduling in Cloud Environment Using Nested PSO Framework, 21 Aug, 2015.
- [11] Li, X., Xu, J., & Yang, Y. (2015). 'A chaotic particle swarm optimization-based heuristic for market-oriented task-level scheduling in cloud workflow systems'. *Computational Intelligence and Neuroscience*, 2015, 81-92.
- [12] Tawfeek, M., El-Sisi, A., Keshk, A. E., &Torkey, F. (2013, November). Cloud task scheduling based on ant colony optimization ", *The International Arab Journal of Information Technology*, Vol. 12, No. 2, March 2015 pp.129-137.
- [13] Anuradha, M., &Selvakumar, S. ACO Based Task Scheduling Algorithm for Hybrid Cloud. *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE) ISSN: 0976-1353 Volume 13 Issue 1 –MARCH 2015*
- [14] Shengjun, X., Mengying, L., Xiaolong, X., Jingyi, C. (2014). 'An ACO-LB Algorithm for Task Scheduling in the Cloud Environment'. *Journal of Software*. 9, 2, 466-473
- [15] Keshk, A. E., El-Sisi, A., Tawfeek, M. A., &Torkey, F. A. (2013). Intelligent strategy of task scheduling in cloud computing for load balancing. *International Journal of Emerging Trends and Technology in Computer Science*, 2(6), 12-22.
- [16] Kokilavani, T., &Amalarethinam, D. D. G. (2012). Memory Constrained Ant Colony System For Task Scheduling In Grid Computing. *International Journal of Grid Computing & Applications (IJGCA) Vol, 3*.
- [17] Kianpisheh, S., Charkari, N. M., &Kargahi, M. (2016). Reliability-driven scheduling of time/cost-constrained grid workflows. *Future Generation Computer Systems*, 55, 1-16.
- [18] Linda J and Ananthanarayana V.S., "Energy-aware Task scheduling in cloud using Ant-colony Optimization", *Proceedings of International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA-14)*
- [19] Karaboga, D., &Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. *Applied soft computing*, 8(1), 687-697.
- [20] Navimipour, N. J. Task scheduling in the Cloud Environments based on an Artificial Bee Colony Algorithm. *Proceedings of 2015 International Conference on Image Processing, Production and Computer Science (ICIPCS'2015) Istanbul (Turkey), June 3-4, 2015 pp. 38-44*
- [21] Monika Rathore, Sarvesh Rai, NavdeepSaluja, "Load Balancing of Virtual Machine Using Honey Bee Galvanizing Algorithm in Cloud", *International Journal of Computer Science and Information Technologies*, Vol. 6 (4) , 2015, 4128-4132
- [22] Kumar, R. S., &Gunasekaran, S. (2014). Improving Task Scheduling in Large Scale Cloud Computing Environment using Artificial Bee Colony Algorithm. *International Journal of Computer Applications*, 103(5).

- [23] Gao, K. Z., Suganthan, P. N., Chua, T. J., Chong, C. S., Cai, T. X., & Pan, Q. K. (2015). A two-stage artificial bee colony algorithm scheduling flexible job-shop scheduling problem with new job insertion. *Expert Systems with Applications*, 42(21), 7652-7663.
- [24] Li, J. Q., & Pan, Q. K. (2015). Solving the large-scale hybrid flow shop scheduling problem with limited buffers by a hybrid artificial bee colony algorithm. *Information Sciences*, 316, 487-502.
- [25] Muthiah, A., & Rajkumar, R. (2014). A Comparison of Artificial Bee Colony algorithm and Genetic Algorithm to Minimize the Makespan for Job Shop Scheduling. *Procedia Engineering*, 97, 1745-1754.
- [26] X. S. Yang, A New Metaheuristic Bat-Inspired Algorithm, in: *Nature Inspired Cooperative Strategies for Optimization (NISCO 2010)* (Eds. J. R. Gonzalez et al.), *Studies in Computational Intelligence*, Springer Berlin, 284, Springer, 65-74 (2010).
- [27] Gandomi, A. H., & Yang, X. S. (2014). Chaotic bat algorithm. *Journal of Computational Science*, 5(2), 224-232.
- [28] Taheri, J., Lee, Y. C., Zomaya, A. Y., & Siegel, H. J. (2013). A Bee Colony based optimization approach for simultaneous job scheduling and data replication in grid environments. *Computers & Operations Research*, 40(6), 1564-1578.
- [29] Zhang, W., Tan, S., Lu, Q., Liu, X., & Gong, W. (2015). A Genetic-Algorithm-Based Approach for Task Migration in Pervasive Clouds. *International Journal of Distributed Sensor Networks*.
- [30] Yu, J., & Buyya, R. (2006). Scheduling scientific workflow applications with deadline and budget constraints using genetic algorithms. *Scientific Programming*, 14(3-4), 217-230.
- [31] Joanna Kolodziej a, Samee Ullah Khan, "Multi-level hierarchic genetic-based scheduling of independent jobs in dynamic heterogeneous grid environment"
- [32] Xu, Y., Li, K., Hu, J., & Li, K. (2014). A genetic algorithm for task scheduling on heterogeneous computing systems using multiple priority queues. *Information Sciences*, 270, 255-287.
- [33] Dasgupta, K., Mandal, B., Dutta, P., Mandal, J. K., & Dam, S. (2013). A genetic algorithm (ga) based load balancing strategy for cloud computing. *Procedia Technology*, 10, 340-347.
- [34] Kumar, P., Verma, A. (2012). 'Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm'. *International Journal of Advanced Research in Computer Science and Software Engineering*. 2, 5, 111-114
- [35] Li, J., Feng, L., & Fang, S. (2014). An greedy-based job scheduling algorithm in cloud computing. *Journal of Software*, 9(4), 921-925.
- [36] Selvi, S., Maheswari, R., & Kalaavathi, B. Deadline-Cost Based Job Scheduling Using Greedy Approach in a Multi-Layer Environment.
- [37] Singh, L. S. V., & Ahmed, J. A GREEDY ALGORITHM FOR TASK SCHEDULING & RESOURCE ALLOCATION PROBLEMS IN CLOUD COMPUTING. *International Journal of Research & Development in Technology and Management Science-Kailash*, 21(1).
- [38] Parikh, S., & Sinha, R. (2013). Double Level Priority based Optimization Algorithm for Task Scheduling in Cloud Computing. *International Journal of Computer Applications*, 62(20).
- [39] Zhou, Z., & HU, Z. (2014). Task scheduling algorithm based on greedy strategy in cloud computing. *Open Cybernetics & Systemics Journal*, 8, 111-114.
- [40] Gu, J., Hu, J., Zhao, T., & Sun, G. (2012). A new resource scheduling strategy based on genetic algorithm in cloud computing environment. *Journal of Computers*, 7(1), 42-52.
- [41] Li, X., Jiang, T., & Ruiz, R. (2016). Heuristics for periodical batch job scheduling in a MapReduce computing framework. *Information Sciences*, 326, 119-133.
- [42] Bansal, N., Maurya, A., Kumar, T., Singh, M., & Bansal, S. (2015). Cost performance of QoS Driven task scheduling in cloud computing. *Procedia Computer Science*, 57, 126-130.
- [43] Thomas, A., Krishnalal, G., & Raj, V. J. (2015). Credit Based Scheduling Algorithm in Cloud Computing Environment. *Procedia Computer Science*, 46, 913-920.
- [44] Lee, Y. C., Han, H., Zomaya, A. Y., & Yousif, M. (2015). Resource-efficient workflow scheduling in clouds. *Knowledge-Based Systems*, 80, 153-162.
- [45] Alkhanak, E. N., Lee, S. P., & Khan, S. U. R. (2015). Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems*.

- [46] Xu, H., & Yang, B. (2015). An incentive-based heuristic job scheduling algorithm for utility grids. *Future Generation Computer Systems*, 49, 1-7.
- [47] Kianpisheh, S., Charkari, N. M., & Kargahi, M. (2016). Reliability-driven scheduling of time/cost-constrained grid workflows. *Future Generation Computer Systems*, 55, 1-16.
- [48] Huang, Y., Bessis, N., Norrington, P., Kuonen, P., & Hirsbrunner, B. (2013). Exploring decentralized dynamic scheduling for grids and clouds using the community-aware scheduling algorithm. *Future Generation Computer Systems*, 29(1), 402-415.
- [49] Durillo, J. J., Nae, V., & Prodan, R. (2014). Multi-objective energy-efficient workflow scheduling using list-based heuristics. *Future Generation Computer Systems*, 36, 221-236.
- [50] Ghafarian, T., & Javadi, B. (2014). Cloud-aware data intensive workflow scheduling on volunteer computing systems. *Future Generation Computer Systems*.
- [51] Vasile, M. A., Pop, F., Tutueanu, R. I., Cristea, V., & Kołodziej, J. (2014). Resource-aware Hybrid scheduling algorithm in heterogeneous distributed computing. *Future Generation Computer Systems*.

AUTHORS

Jyoti Thaman is working as Assistant Professor in Department of Computer Science and Engineering, in M. M. University Mullana, Ambala. She obtained her Masters from Kurukshetra University, Haryana, India, in 2007. Her interest lies in Cloud and Grid Computing. She is Ph. D Candidate from the Department of Computer Science and Engineering, M. M. University, Ambala, India. She has published three research papers.

Dr. Manpreet Singh is working as Professor and Principal, M M Group of Institutions, M. M. University, Sadopur, Ambala, India. He obtained his doctorate from Kurukshetra University in 2010. He completed his M. Tech and B. Tech. from Kurukshetra University, India. He has teaching and research experience of more than 16 years. He has published more than 40 research papers in leading journals and conferences. His research areas are Grid Computing, Cloud Computing and Ad hoc Networks.