International Journal of
Emerging Research in Management &Technology
ISSN: 2278-9359 (Volume-6, Issue-8)

Research Article | **August 2017**

Review on Multiobjective Task Scheduling in Cloud Computing using Nature Inspired Algorithms

Shikha Chaudhary
Department of CSE,
RIET, Jaipur, Rajasthan, India

Saroj Hiranwal
Department of CSE,
RIET, Jaipur, Rajasthan, India

C. P. Gupta
Department of CSE,
UCE, Kota, Rajasthan, India

Abstract—

In cloud computing huge pool of resources are available and shared through internet. The scheduling is a core technique which determines the performance of a cloud computing system. The goal of scheduling is to allocate task to appropriate machine to achieve one or more QOS. To find the suitable resource among pool of resources to achieve the goal is an NP Complete problem. A new class of algorithm called nature inspired algorithm came into existence to find optimal solution. In this paper we provide a survey as well as a comparative analysis of various existing nature inspired scheduling algorithms which are based on genetic algorithm and ant colony optimization algorithm.

Keywords— cloud computing, scheduling, ant colony optimization, genetic algorithm, nature inspired algorithm

I. INTRODUCTION

Cloud Computing is developing as a auspicious paradigm for providing computing resources like networks, servers, storage, applications as a service over the internet. Resources can be promptly allocated and released with slight management effort. It is an approach where computing is delivered as a service rather than a product. Cloud Computing provides everything as a service (XaaS) like Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).In SaaS , applications are provided to user through client devices like web browser and user need not to manage anything like storage, server, network, operating system etc eg: Google apps .In PaaS , the platform at which software can be developed and deployed ,provided to user and user need not to manage underlying hardware eg: Google App Engine. In IaaS , computing resources like servers, processing power, storage, and networking are provided to user as a service [1].

Scheduling is to allocate task to appropriate machine to achieve some objectives i.e. it determines on which machine which task should be executed. Unlike traditional scheduling where the tasks are directly mapped to resources at one level , resources in cloud are scheduled at two level i.e. physical level and VM level (Fig 1).There are mainly two type of task scheduling in cloud computing: static scheduling and dynamic scheduling. In static task scheduling information of task is known before execution like execution time whereas in dynamic task scheduling, information of task is not known before execution [2]. In cloud environment to execute a task a user request for a computing resource which is allocate by cloud provider after finding the appropriate resource among existing as shown in figure 1. Tasks which are submitted for execution by users may have different requirements like execution time, memory space, cost, data traffic, response time, etc. In addition, the resources which are involved in cloud computing may be diverse and geographically dispersed. There are different environments in cloud: single cloud environment and multi-cloud environment.

Scheduling in single-cloud environments is where virtual machines (VM) are scheduled within an infrastructure provider that can have multiple data centers which are physically distributed. In single-cloud scenario characteristic of resources, dynamicity, cost etc. are usually focused as optimization metrics. Whereas multi-cloud environments include a cloud infrastructure that unburdens its workload to another infrastructure and a cloud user who deploys and manages VMs across multiple cloud infrastructures gaining, improving service availability and fault-tolerance, etc.

These characteristics raise challenges in forming task scheduling in cloud. Many researchers have created various algorithms by considering one of these parameters and by ignoring others.

Scheduling process in cloud can be categorized into three stages namely–

a. Resource discovering and filtering – Resource request is made by cloud user and submitted to service provider, service provider searches the suitable resources, locate them .
b. Resource selection –Resource is selected on the basis of task and resource selection parameters.
c. Task submission -Task is submitted to selected resource.

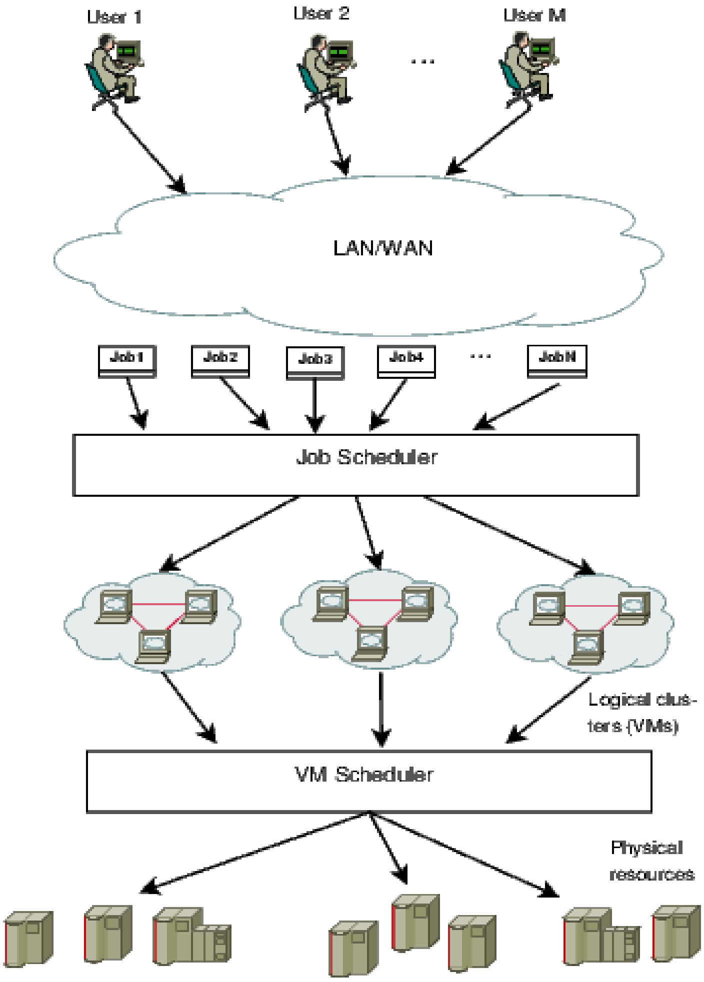

Fig 1: View of cloud

II. SCHEDULING CRITERIA

There are various objective functions or matrices for selecting a scheduling algorithm. The criteria differ with respect to service provider and user. Service provider wants to gain revenue, maximize resource utilization with minimal efforts. Whereas user wants his job to be executed with minimal cost in minimum time [3].

A. Cloud User Preferred

1) Makespan: it tells about the finishing time of last task. The makespan should be minimum which shows the fastest execution time of a task.
2) Cost: it is the sum of the amount paid by the user to provider for using individual resource.
3) Waiting time: the time spent by a task in ready queue to get a chance for execution.
4) Turn around time: Time taken by a task to complete its execution after its submission i.e. the sum of waiting time and execution time of a task.
5) Tardiness: the delay in execution of a task i.e. difference between finishing time and deadline of the task. For an optimal scheduling the tardiness should be zero which shows no delay in execution.
6) Fairness: this shows that all tasks are getting equal opportunity of execution.
7) Response time: time taken by a system to start responding (first response) after submission of a task.

B. Cloud provider preferred

1) Resource utilization: the resources should be fully utilized by keeping them as busy as possible to gain the maximum profit.
2) Throughput: his represents the number of task completed in a per time unit.
3) Predictability: this represents the consistency in the response times of task .Unpredictable response time may degrade the performance of system.
4) Priority: To give preference to a task to finish it as earliest. Priority can be given on the basis of arrival time, execution time, deadline etc. Resources are provided to higher priority task to complete the execution.
5) Load balancing: distribution of load among all the computing resources.
6) Deadline: the time till which a task should be completed.
7) Energy efficiency: Reducing the amount of energy used to provide any solution or service.

III. STUDY OF EXISTING SCHEDULING ALGORITHMS

Scheduling is dispatching of task to the computing resource among the pool of resource by applying certain rules under different circumstances. There is no standard algorithm which can schedule the task efficiently under certain circumstances. Various traditional scheduling algorithms like First come first served (FCFS), Round Robin, Fairness, Max-Min, Min-Min, Priority, and Shortest Job First exist. The task scheduling is a NP-Complete problem because the aim of scheduling is to map the tasks to unlimited available computing resources by meeting objective functions or scheduling criteria. No such algorithm exists which can provide optimal solution in polynomial time. In this paper we will study about some of existing scheduling approaches and comparison is shown in table 1.

A. Nature Inspired Algorithm

Optimization is to select the best solution among the given set of options which maximizes or minimizes the objective function. There exist various real world problems where complicated systems are modelled with multi objective functions. Sometime these problems are difficult to solve or can be time consuming. Therefore, a new class of algorithm came into existence which is called as nature inspired algorithms. Researchers analysed that several system in nature are exist which are similar to optimization problem. So the various algorithms have been introduced by studying the nature [4]. Natures inspired algorithms can be divided into two categories like swarm based optimization and evolutionary algorithms, which are further categorized as shown in figure 2.In this paper task scheduling algorithms based on one algorithm of each category are analysed.

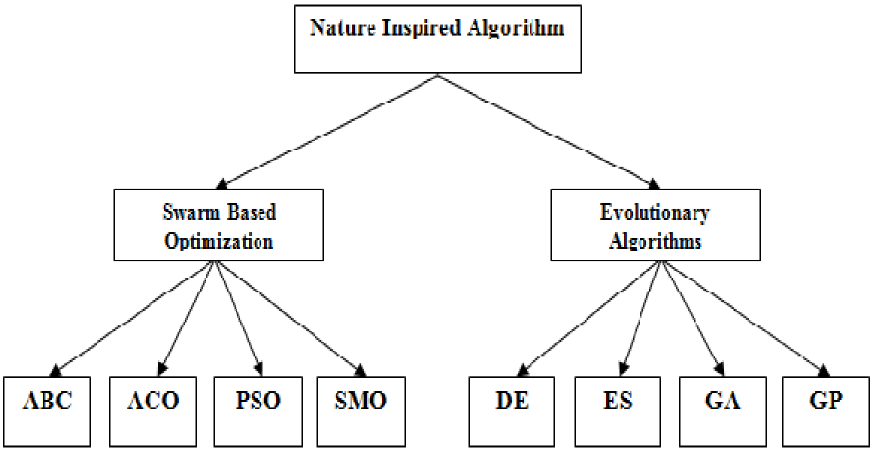

Fig 2: Nature Inspired Algorithms

1) Genetic Algorithm Based Scheduling Algorithms:

GA was first introduced by Holland in 1975 and represents a population based optimization method based on evolution process perceived in nature. In GA, each chromosome which is an individual in the population represents a probable solution to a problem and is poised of a string of genes. The initial population is taken randomly at the beginning of algorithm. A fitness function is calculated to check the appropriateness of the chromosome for the environment. Chromosomes are selected on the basis of fitness value. Crossover and mutation operations are accomplished on selected chromosomes to generate offspring for the new population. The fitness function evaluates the quality of each offspring. The process is repeated until sufficient offspring are created [5,6].

Algorithm of GA

- i. randomly initialize population(t)
- ii. determine fitness of population(t)
- iii. repeat
  - a. select parents from population(t)
  - b. perform crossover on parents creating population(t+1)
  - c. perform mutation of population(t+1)
  - d. determine fitness of population(t+1)
- iv. until best individual is good enough

Ge Y, Wei G [7] proposed a GA based task scheduler which provides load balancing. Tasks are scheduled using sliding window technique where the window size is fixed and the task which comes in sliding window will get scheduled. This algorithm works on dynamic scheduling so to predict the execution time of individual task a Kernel canonical Correlation Analysis (KCCA) technique is used. After that the GA is applied to map the task to VM according to the state of the system which is updating iteratively using GA. Chang-Tian Jiong [8] proposed energy-aware Genetic Algorithms tasks scheduling by taking makespan and energy as the scheduling criteria. They applied Dynamic Voltage Scaling (DVS) to minimize energy consumption and proposed two algorithms. These two algorithms Energy consumption Time Unify Genetic Algorithm (ETU_GA) and Energy consumption Time Double fitness Genetic Algorithm (ETDF_GA) are used to define the fitness function and to select individuals. Pop et. Al [9] proposed a GA based approach where the selection phase is based upon a reputation ranking given by the previous consumers who used the resources for execution. Reputation is based upon execution time, transmission time, capacity, price. Which resultant the good load balancing, maximize profit, reduce cost. Wang, Liu, Chen, Xu, Xi and Dai [10] proposed a task scheduling algorithm based on genetic algorithm with the aim of minimizing make-span and even distribution of load between virtual machines. The authors used greedy algorithm to initialize the population and two fitness functions are applied for selection strategy. Both fitness functions are reciprocal of inter load variance and of task completion time respectively. Selection is based on the fitness ratio, higher the ratio than higher the chance to be get selected. They used adaptive probabilities for crossover and mutation instead of using fixed values. Kun-lun, Jun, Jian, Qing-yun [11] proposed an improved GEP algorithm with double fitness functions (DF-GEP). The algorithm is based on Map/Reduce programming model. A modified ETCC matrix is proposed which consist the task completion time as well as operating cost of the tasks while ordinary matrix consists only task completion time. The algorithm has improved the encoding and decoding strategy as compared to GEP algorithm and also applies double fitness functions .This algorithm minimizes the total task completion time as compared to GEP algorithm and also reduces the running cost of tasks.Chun-Yan LIU, Cheng-Ming ZOU, Pei WU [12] integrated the Genetic algorithm (GA) and Ant Colony Optimization algorithm (ACO) to get the optimal solution, where it uses the positive feedback of ant colony optimization (ACO) and the strong searching ability of genetic algorithm. The initial pheromone is calculated using GA algorithm and finally gets the optimal scheduling using ACO. This algorithm searches the resource fast for the faster execution. Verma, Kaushal [13] proposed BCHGA to schedule applications to cloud resources that minimizes the execution cost while meeting the Budget for delivering the result. Each workflow's task is assigned priority. To increase the population diversity, these priorities are then used to create the initial population of BCHGA Work on pay-as-you-use pricing model like Amazon. Singh, Kalra [14] proposed a modified GA to minimize makespan. In this approach initial population is generated by Enhanced Max-Min algorithm (task having average execution time is assigned to slowest resource) rather than random approach. On this initial population crossover and mutation is applied and the new generated offspring/ schedule is added to the population. Schedule which gives the minimum makespan from the whole population is selected. Kar, Parida, Das [15] proposed a energy-aware Genetic Algorithms for task scheduling. It calculated energy value of each task using Max-Min algorithm. Genetic Algorithm is applied to allocate tasks to Virtual machines in the host. Meena et. al [16] proposed a cost effective genetic algorithm for encoding, population initialization, crossover, and mutation operators of the Genetic Algorithm. It reduces the execution and cost of the workflow by meeting the deadline. Bei, Jun proposed [17] an improvement to standard genetic algorithm; it takes into account the balance between global and local search ability. The min-min and max-min algorithm are used for the population initialization. Fitness function is calculated by considering time and cost.to improve the search efficiency, adaptive crossover and mutation operations are adopted so that poor individuals can also be accepted with a certain probability then the population diversity can be maintained and the local optimum can also be avoided. Fu, Lu, Wu [18] proposed a time-clustering-based genetic algorithm method after analyzing the energy consumption characteristics and the SLA time constraint of surveillance tasks in Cloud Video Surveillance (CVS) systems. Tasks with close access time periods are gathered as clusters using the time clustering method. Therefore, surveillance tasks in a day are divided into clusters on the basis of different access time periods and scheduled separately accordingly. A genetic scheduling algorithm for all clusters in the CVS data center is designed. The algorithm improves the resource utilization and reduces energy consumption.

2) Ant Colony Optimization Based Scheduling Algorithms:

This novel approach was introduced by Dorigo in 1992 and was originally called ant system. It is based on the behavior of real ants finding the shortest path between their colonies and a source of food .While walking through their colony and the food source; ants leave pheromones on the ways they move. The pheromone intensity on the passages increases with the number of ants passing through and drops with the evaporation of pheromone. After some time smaller paths have more pheromone and thus, pheromone intensity helps ants to identify smaller paths to the food source [19].

ACO Algorithm:

1. Initialize pheromone
2. Do While (Stopping Criteria Not Satisfied) – Cycle Loop
    i. Do Until (Each Ant Completes a Tour) – Tour Loop
    ii. Local pheromone Update
    iii. End Do
    iv. Analyze Tours
    v. Global pheromone Update
3. End Do

Li, Xu, Zhao, Dong, Wang [20] presented Load Balancing Ant Colony Optimization (LBACO) algorithm, for scheduling of independent tasks with the aim of minimizing makespan and even load across all virtual machines. They have calculated the degree of imbalance to measure the imbalance among virtual machines. Their experimental results show that LBACO has reduced the average makespan in comparison with FCFS algorithm. Lu and Gu[21] proposed a dynamic load balancing strategy for cloud based on ACO. They recognized overloaded resource by finding out the virtual machines load on the basis of CPU usage, memory usage and bandwidth, if values are higher than the threshold values in real time then the resource is overloaded.

Table 1. Comparisons of various scheduling algorithms

Paper	Nature of Task	Optimization parameters	Tool
GA-based task scheduler for the cloud computing systems [7]	Independent	makespan	CloudSim
Energy-aware Genetic Algorithms for Task Scheduling in Cloud Computing[8]	Independent	Makespan, energy	CloudSim
Reputation guided Genetic Scheduling Algorithm for Independent Tasks in Inter-Clouds Environments[9]	Independent	Load balancing, cost	--
Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing[10]	Independent	Makespan, load balancing	matlab
Improved GEP Algorithm for Task Scheduling in Cloud Computing [11]	Independent	Makespan, cost	Matlab , GridSim
A task scheduling algorithm based on genetic algorithm and ant colony optimization[12]	Independent	Running time	CloudSim
budget constrained priority based genetic algorithm for workflow scheduling in cloud[13]	Workflow	Cost	Java
Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm[14]	Independent	Makespan	CloudSim
Energy Aware Scheduling using Genetic Algorithm in Cloud Data Centers[15]	Independent	Energy, makespan, cost	CloudSim
Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud under Deadline Constraint[16]	Workflow	Execution time, cost	CloudSim
Load Balancing Task Scheduling based on Multi-Population Genetic Algorithm in Cloud Computing[17]	Independent	Load balancing, cost	MATLAB
A Task Scheduling Method for Energy-Efficient Cloud Video Surveillance System Using A Time-Clustering-Based Genetic Algorithm[18]	Independent	Resouce utilization, energy	CloudSim
Cloud task scheduling based on load balancing ant colony optimization[20]	Independent	makespan	CloudSim
A load-adapative cloud resource scheduling model based on ant colony algorithm[21]	Independent	Load balancin	Physical Implementation
Energy-aware ant colony based workload placement in clouds[22]	Independent	Resouce utilization, energy	CloudSim
Load balancing of nodes in cloud using ant colony optimization[23]	Independent	Load balancing	--
Cloud task scheduling based on ant colony optimization[24]	Independent	Makespan	CloudSim
PACO: A Period ACO_based Scheduling Algorithm in Cloud Computing[25]	Independent	Load balkancing, makespan	CloudSim
Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization[26]	Independent	Energy	--
SAACO: A Self Adaptive Ant Colony Optimization in Cloud Computing[27]	Independent	Makespan, load balancing	CloudSim

The load from these machines is moved to nearest idle nodes using ACO algorithm. Nearest idle VM is found using ant colony algorithm and the load is transferred to idle load from the overloaded VM. This approach achieves the goal of better load balancing. An ACO based energy aware approach has been proposed in [22].As energy consumption by data centres is a major issue. A conventional strategy to save energy in data centres is by mapping virtual machines on minimum physical machines. The mapping is considered as Multi-dimensional Bin Packing (MDBP) problem where physical machines are bins and virtual machines represent the objects to be assigned. Kumar et al.[23] modified ant colony optimization algorithm for balancing load in cloud. They introduced the concept of foraging and trailing pheromones to search overloaded and under loaded resource. In their approach ants works on a common result set and continuously update it rather than their individual result set. By using this, a solution set is gradually made and continuously improved. Tawfeek et al. [24] have taken minimization of makespan as the objective function. They have taken a constraint of visiting each Virtual Machine (VM) once for each ant and heuristic function is based on expected execution time and transfer time of task on VM. The algorithm is simulated in Cloudsim with the number of tasks varying from 100 to 1000. ACO is compared with Round Robin and FCFS algo-rithms and experimental results prove that with the increase in number of tasks, ACO takes less time than RR and FCFS. Weifeng Sun,Ning Zhang, Haotian Wang, Wenjuan Yin, Tie Qiu [25] proposed an algorithm Period ACO based Scheduling Algorithm(PACO). In this approach when a task is assigned onto a resource, the pheromone intensity of the selected resource will be reduced which reduces the chances of selection of the resource by other tasks. If there are too many tasks to assign they will be assigned in different period to a resource which can accept as many tasks as it can according to its processing capacity. And if all the resources' pheromone intensity reduced to the minimum value, all the tasks that are not assigned will select the resource which has the largest innate pheromone intensity according to among them. It provides better load balancing and minimizes makespan as compared to Min-Min algorithm. Liu et. al [26] proposed the algorithm to solve the virtual machine placement problem to effectively utilize the physical resources and to reduce the number of running physical servers. In this algorithm ACO approach tried to reduce the number of physical servers. Instead of deploying the pheromone between virtual machines and physical server, it deployed between the pair of virtual machines to track the past desirability of placing them in the same physical machine. Sun, Ji, Sun, Zhang, Hu [27] proposed Self Adaptive Ant Colony Optimization (SAACO) algorithms where they integrated particle swarm optimization (PSO) and ACO. The initial value of pheromone is calculated using PSO to make the parameters of ACO to be self-adaptive. They used makespan and load balancing as optimization metrics and compared the algorithm with PACO and Min-Min algorithm, experimental results shows that SAACO is better in performance in terms of makespan and load balancing than PACO and Min-Min.

IV. CONCLUSION

In this report, various task scheduling algorithms using nature inspired algorithm in cloud computing environment have been reviewed. However, Nature Inspired techniques are usually slower than deterministic algorithms and the generated solutions may not be optimal. Thus most of the research done is toward improving the convergence speed and quality of the solution. Different scheduling algorithms have different optimization criteria. Most of the researches that have been reviewed were towards minimizing the total completion time and improving the resource utilization in different cloud environments whereas some of them were focusing on cost, energy etc.

REFERENCES

[1] Barrie Sosinsky, "Defining Cloud Computing" *in Cloud Computing Bible*, Indianapolis, Indiana: Wiley Publishing; Inc. 2011

[2] Y.Chawla, M.Bhonsle, "A Study on Scheduling Methods in Cloud Computing", *International Journal of Emerging Trends & Technology in Computer Science*, Volume 1, Issue 3, pp. 12-17, September –October, 2012.

[3] Xhafa F, Abraham A., "Computational models and heuristic methods for Grid scheduling problems", *Futur Gener Comput Syst*, Volume 26, Issue 4, pp. 608–621, April, 2010.

[4] Xin-She Yang, Nature-Inspired Optimization Algorithms, ISBN: 978-0-12-416743-8

[5] Moraga RJ, DePuy GW, Whitehouse GE., Metaheuristics: a solution methodology for optimization problems. Handb Ind Optim Probl Handb Ind Syst Eng AB Badiru 2006.

[6] Pop F, Dobre C, Cristea V., "Genetic algorithm for DAG scheduling in grid environments", in 5th IEEE int conf intell comput commun process; 2009, pp. 299–305.

[7] Ge Y, Wei G., "GA-based task scheduler for the cloud computing systems", int conf web inf syst min, vol. 2; 2010. p. 181–186. .

[8] Ying Changtian, Yu Jiong, "Energy-aware Genetic Algorithms for Task Scheduling in Cloud Computing", *in proceeding of Seventh China Grid Annual Conference*, 2012.

[9] Florin Pop, Valentin Cristea , Nik Bessis, Stelios Sotiriadis, "Reputation guided Genetic Scheduling Algorithm for Independent Tasks in Inter-Clouds Environments", in proceeding of 27th International Conference on Advanced Information Networking and Applications Workshops, 2013.

[10] Tingting Wang, Zhaobin Liu, Yi Chen, Yujie Xu, Xiaoming Dai, "Load Balancing Task Scheduling based on Genetic Algorithm in Cloud Computing", in proceeding of IEEE 12th International Conference on Dependable, Autonomic and Secure Computing, pp: 146-152, August 2014.

[11] LI Kun-lun; Wang Jun; Song Jian; Dong Qing-yun, "Improved GEP Algorithm for Task Scheduling in Cloud Computing" , in proceeding of Second International Conference on Advanced Cloud and Big Data, pp:93-99, Nov 2014.

[12] Chun-Yan LIU, Cheng-Ming ZOU, Pei WU, "A task scheduling algorithm based on genetic algorithm and ant colony optimization in cloud computing", in proceeding of 13th International Symposium on Distributed Computing and Applications to Business, Engineering and Science, ISBN: 978-1-4799-4168-1, January 2015.

[13] Amandeep Verma , Sakshi Kaushal, "Budget constrained priority based genetic algorithm for workflow scheduling in cloud" , in proceedings of Communication and Computing, pp.216-222,2013.

[14] Shekhar Singh, Mala Kalra, "Scheduling of Independent Tasks in Cloud Computing Using Modified Genetic Algorithm", in proceedings of Sixth International Conference on Computational Intelligence and Communication Networks, 2016.

[15] Ipsita Kar, R.N. Ramakant Parida, Himansu Das, " Energy Aware Scheduling using Genetic Algorithm in Cloud Data Centers", in proceedings of International Conference on Electrical, Electronics, and Optimization Techniques(ICEEOT)–2016.

[16] Jasraj Meena, Malay Kumar, Manu Vardhan, "Cost Effective Genetic Algorithm for Workflow Scheduling in Cloud under Deadline Constraint", IEEE Access. pp. 5065 – 5082, Aug 2016.

[17] WANG Bei, LI Jun, "Load Balancing Task Scheduling based on Multi-Population Genetic Algorithm in Cloud Computing", in proceedings of the 35th Chinese Control Conference, July 27-29, 2016, Chengdu, China.

[18] Dongping Fu, Yonghua Xiong, Chengda Lu, Min Wu, "A Task Scheduling Method for Energy-Efficient Cloud Video Surveillance System Using A Time-Clustering-Based Genetic Algorithm", in proceedings of 22nd International Conference on Parallel and Distributed Systems, IEEE, Dec 2016.
[19] Dorigo M, Stu¨tzle T., Ant colony optimization, MIT Press; 2004.
[20] Li K, Xu G, Zhao G, Dong Y, Wang D., "Cloud task scheduling based on load balancing ant colony optimization", in proceedings of Sixth Annual China grid Conf , Aug 2011.
[21] Lu X, Gu Z., "A load-adapative cloud resource scheduling model based on ant colony algorithm", in proceeding of IEEE int conf cloud comput intell sys, 2011. pp. 296–300.
[22] Feller E, Rilling L, Morin C., "Energy-aware ant colony based workload placement in clouds.", in proceeding of 12th IEEE/ACM int conf grid comput, 2011, pp. 26–33.
[23] Nishant K, Sharma P, Krishna V, Gupta C, Singh KP, Nitin," Load balancing of nodes in cloud using ant colony optimization", in proceeding of 14th int conf Computer Modelling and Simulation (UKSim), IEEE, 2012, pp.3–8.
[24] Tawfeek MA, El-Sisi A, Keshk AE, Torkey FA, "Cloud task scheduling based on ant colony optimization", in proceedings of 8th int conf comput eng syst, IEEE, 2013.
[25] Weifeng Sun Ning Zhang, Haotian Wang, Wenjuan Yin, Tie Qiu, "PACO: A Period ACO_based Scheduling Algorithm in Cloud Computing", in proceedings of International Conference on Cloud Computing and Big Data.
[26] Liu X, Zhan Z, Du K, Chen W, "Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization", In: Proc conf genet evol comput.,ACM,2014.
[27] Weifeng Sun, Zhenxing Ji, Jianli Sun, Ning Zhang, Yan Hu, "SAACO: A Self Adaptive Ant Colony Optimization in Cloud Computing", in proceedings of Fifth International Conference on Big Data and Cloud Computing, IEEE, 2015.