Engineering Universe for scientific Research and management
ISSN (Online): 2319-3069
Vol. XV Issue V
May 2023

A Review on Energy Efficiency Scheduling Approaches in Cloud Computing

Dr. Narendra Sharma[1], Farheen Sultana[2]
Assistant Professor, Sri Satya Sai University of technology and Medical Sciences, Sehore, India[1]
Research Scholar, Sri Satya Sai University of technology and Medical Sciences, Sehore, India[2]

Abstract

Over the past ten years, cloud computing has emerged as the platform of choice for handling problems and requests on the Internet. Without the need for any client setup, cloud computing offers fantastic opportunity for running efficient scientific workflows. It makes practically limitless resources accessible, allowing them to be gathered, organised, and applied as needed. Resource scheduling is essential to the efficient distribution of resources across all tasks in the cloud environment. However, in order to offer an effective scheduling algorithm, numerous problems must also be taken into account. The application of objectives such scheduling cost, load balancing, make span time, security awareness, energy consumption, dependability, service level agreement maintenance, etc. must be improved by an efficient scheduling algorithm. Many cutting-edge scheduling methods based on hybrid, heuristic, and meta-heuristic approaches have been presented to meet the aforementioned aims. In this study, existing algorithms were examined in light of the scheduling purpose and tactics. We compare the results offered by the various tactics that are now in use. In order to provide insight into future research and unresolved issues, we emphasize the downsides.

Keywords: *Scheduling, Cloud Computing, Energy, State-of-the-art, Load balancing*

1. Introduction

Cloud computing is internet based computing, whereby shared servers provides software, resources and data to the users and other services on demand. In case of providing such services, scheduling the request is most important. The services rendered by cloud to users grows day-by-day by an enormous amount. So, cloud service provider need to provide the services to the users request effectively and efficiently. Job scheduling is one of the major activities performed in all the computing environment. Cloud computing is one the upcoming and current trend latest technology which is developed drastically. To efficiently and effectively increase the working of cloud computing environment, job scheduling is one the tasks performed in order to gain maximum profit. The main goal of scheduling algorithms in distributed systems is spreading the load on processors and while minimizing their utilization. Job scheduling one of the most famous optimization problem, plays a key role to improve reliable and flexible systems. The main purpose is to schedule jobs according with adaptable time, which involves finding out a proper sequence in which jobs can be executed under transaction logic constraints. There are major two categories of scheduling algorithm.1).Dynamic Scheduling and 2).Static Scheduling algorithm. Both have their own limitation and advantage. Static scheduling algorithm have lower performance than Dynamic scheduling algorithm but has a lot of overhead compare to it.

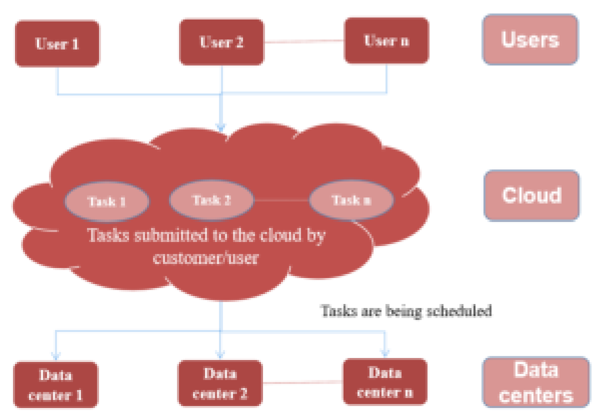

Figure 1. Task scheduling in cloud

2. Scheduling Techniques

There has been various types of scheduling algorithms present in shared computing environment. Most of them can be applied in the cloud computing environment with suitable information. The major advantage of job scheduling algorithm is to achieve a best system throughput and high performance computing .Traditional job scheduling algorithms are not able to provide scheduling in the cloud environment. According to a simple allocation, job scheduling algorithms in cloud computing can be categorized into two major groups:

Engineering Universe for scientific Research and management
ISSN (Online): 2319-3069
Vol. XV Issue V
May 2023

Online Mode Heuristic Algorithm (OMHA) and Batch Mode Heuristic Algorithm(BMHA) .In BMHA, when jobs are arrive in the system they are queued and collected into a list. The scheduling algorithm will start after fixed period of time. The major example of BMHA based scheduling algorithms are: First Come First Served Algorithm (FCFS), Round Robin (RR), Max-Min and Min-Min algorithm. By OMHA, when jobs are arrive in the system and they are scheduled. Since the cloud computing is a heterogeneous system and the speed of each processor differs quickly, the OMHA are more suitable for cloud computing environment. Most Fit Task Scheduling algorithm (MFTS) is suitable example of Online Mode Heuristic Algorithm.

2.1 Scheduling Process

Scheduling process in cloud computing can be classified into three stages namely:

1. Resource Analysis and Cleaning: Datacenter broker discovers the resources present in the network system and collects status information related to them.
2. Resource Selection: Target Resource is selected based on certain parameters of resource and task
3. Submit Task: Task is submitted to resource selected.

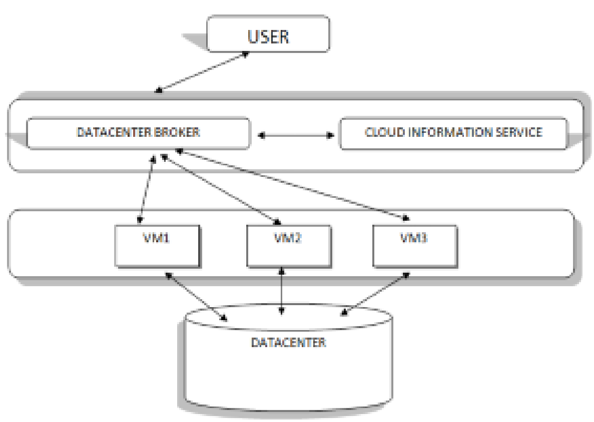

Figure 1: Scheduling in cloud computing

There are various types of scheduling techniques in which some of them are describing below:

2.2 Ant Colony Optimization (ACO) Scheduling algorithm

Ant Colony Optimization (ACO) metaheuristic is inspired by the behavior of real ants finding the shortest path between their colonies and a source of food. This novel approach was introduced by Dorigo in 1992 in his Ph.D. thesis and was originally called ant system. While walking amid 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. As the time goes on, smaller paths draw more pheromone and thus, pheromone intensity helps ants to recognize smaller paths to the food source [7].

ACO methods are useful for solving discrete optimization problems that need to find paths to goals. It has been successfully applied for solving traveling salesman problem, multidimensional knapsack problem, job shop scheduling, quadratic assignment problem, scheduling of tasks in grid and cloud and many more. The first step toward any problem solution using ACO is to map ant system to the given problem.

Chen et al. [1] addressed the time-varying workflow scheduling problem in grids based on ACO approach intended to minimize the total cost in a period while meeting the deadline constraint. For this, integrated heuristic is designed based on the average value of cost heuristics and deadline heuristics. The fitness of a schedule is evaluated by considering its performance in different topologies in a period. ACO can be improved by using knowledge gained from predetermined number of best solutions of previous iterations [2]. The concept of knowledge matrix is integrated with the ACO algorithm. Knowledge matrix is changed by two methods, one is by knowledge depositing rule and other is by knowledge evaporating rule. Knowledge depositing encompasses multiplying the knowledge matrix with a constant and is performed on best schedule found till then whereas knowledge evaporation is done after every iteration.

Wen et al. [3] proposed that ACO algorithm can also be combined with other algorithms such as Particle Swarm Optimization (PSO) to improve its performance. The proposed algorithm not only enhances the convergence speed and improves resource utilization ratio, but also stays away from falling into local optimum solution. *Pacini et al. [4]* addressed the problem of balancing throughput and response time when multiple users are running their scientific experiments on online private cloud. The solution aims to effectively schedule virtual machines on hosts. Throughput is related to number of users effectively served and response time is linked to number of virtual machines allocated.

2.3 GA based scheduling algorithms

GA was first introduced by Holland in 1975 and represents a population based optimization method based on a metaphor of the evolution process observed in nature. In GA, each chromosome (individual in the population) represents a possible solution to a problem and is composed of a string of genes. The initial population is taken randomly to serve as the starting point for the algorithm. A fitness function is defined to check the suitability of the chromosome for the environment.

On the basis of fitness value, chromosomes are selected and crossover and mutation operations are performed on them to produce offsprings for the new population. The fitness function evaluates the quality of each offspring. The process is repeated until sufficient offspring are created [4, 5]. In the literature, different types of representations to encode scheduling solutions for GA are used. Fixed bit string representation [6] is the classical approach for representing solutions in GA. In this approach, solutions are encoded into fixed length binary strings. However there have been many modifications to this approach. The three frequently used representations nowadays are direct representation, permutation based representation and tree representation. In direct representation, chromosomes are vectors of size n, where n is the no of tasks and value of ch[i] represents the resource on which task i is scheduled. Direct representation was used in [7-11]. Permutation based representation uses a 2D vector to represent a chromosome. One dimension represents the resources and other dimension shows the order of tasks on each resource. This representation was applied in. Tree representation has been used in [12] for mapping relationship between VMs and physical machines. The initial population is generated randomly in basic genetic algorithm. To obtain optimal results and increase the convergence speed of the GA, some heuristic approaches can be applied to generate the initial population. Minimum Execution Time (MET) and Min–min heuristic have been used to generate initial population. It is used in Longest Job to Fastest Processor (LJFP) and Smallest Job to Fastest Processor (SJFP) for this purpose. As applied GA to solve workflow scheduling problem, precedence of tasks was also considered while generating initial population. In Best-fit and Round-Robin methods are used to select good candidate resources for tasks. Fitness function is used to calculate fitness value of chromosomes. Fitness function may be based on makespan, flow time or execution cost. Selection operators are further used to select chromosomes to which crossover operators are applied. Roulette wheel strategy and Binary Tournament Selection are some of the commonly used selection procedures. Several crossover operators and mutation operators have been explored in the literature. One-point crossover and Two-point crossover operators have been widely used in performing crossover operation. Simple Swap and Swap and Move are commonly used mutation operators. In crossover, all tasks were selected between two successive points of parent1 and exchanged with location of same tasks of parent. The authors used random gene selection crossover in which some randomly selected genes of two parents are changed by each other to produce new offsprings. For mutation, they randomly selected a gene from a chromosome and replaced its resource with a resource having better failure rate and not overloaded.

Zhu et al. [13] presented a multi-agent genetic algorithm (MAGA) for balancing load between virtual machines. MAGA is a combination of GA and multi-agent techniques which reduces convergence time and improves the quality of optimization results as compared to standard GA. The experimental results prove that it is better to use MAGA than basic GA for cloud environment as it can solve large-scale, high dimensional and dynamic optimization problems with ease. Their algorithm balances both CPU utilization and memory usage among virtual machines.

Wang et al. [14] proposed a task scheduling algorithm based on genetic algorithm with the aim of minimizing makespan and even distribution of load between virtual machines. They used greedy algorithm to initialize the population and their selection strategy is based on fitness ratio. Two types of fitness functions are defined, out of which one is selected randomly in each iteration. They have taken adaptive probabilities of crossover and mutation rather than using fixed values.

Shen and Zhang [15] presented energy aware task scheduling algorithm based on shadow price guided genetic algorithm (SGA). Shadow price in GA is defined as the comparative improvement of chromosome's fitness value with the modification of a gene. They have modified genetic operators using shadow price information to enhance the probability of producing better solutions. Their experimental results expose the average energy saving using SGA over GA with 500 tasks is 5.41E+18.

2.4 PSO based scheduling algorithms

Particle Swarm Optimization (PSO) is an evolutionary computational technique introduced by Kennedy and Eberhart in 1995 motivated by social behavior of the particles. Each particle is allied with position and velocity and moves through a multi-dimensional search space. In each iteration, each particle adjusts its velocity based on its best position and the position of the best particle of the whole population. PSO combines local search methods with global search methods trying to balance exploration and exploitation. PSO has gained popularity due to its simplicity and its usefulness in broad range of applications with low computational cost. The first step of applying PSO to scheduling problem is to encode the problem. A commonly used method is to represent the particle as 1 _ n vector, where n is the no. of tasks and value assigned to each position is the resource index.

Thus the particle represents mapping of resource to a task. A matrix based encoding scheme is presented in which m _ n position matrix represents solutions, where m is the no. of resources and n is no. of tasks. The elements of this matrix can have value either 0 or 1 with the constraint of having single element with value 1 in each column. The concept is, each column represents a job allocation and each row represents allocated jobs to a resource. Velocities are also represented in the form of matrices.

Liu et al. [16] used fuzzy matrices to represent position and velocities of particles. The element in each matrix signifies fuzzy relation between resource and job i.e. the degree of membership that the resource would execute the job in the feasible schedule solution space.

Engineering Universe for scientific Research and management
ISSN (Online): 2319-3069 Vol. XV Issue V
May 2023

The next step in PSO is to generate initial population, which is generally produced. As randomness decreases the probability of the algorithm to converge to best solution, *Abdi et al. [74]* created initial particles based on Shortest Job to Fastest Processor (SJFP) Algorithm, whereas Wu et al. [17] generated initial population using Greedy Randomized Adaptive Search Procedure (GRASP).

Zhang et al. [18] proposed to apply Variable Neighborhood Search (VNS), a local search algorithm, after each iteration of PSO to enhance the exploitation of searching space. *Pooranian et al. [19]* have proposed a combination of PSO and Gravitational Emulation Local Search (GELS) algorithm for independent task scheduling in grid computing. GELS is a local search algorithm used to improve the results obtained after PSO, by avoiding local optima. GELS algorithm checks results obtained from PSO to get the best solution and does not explore the search space randomly. The experimental results show that the PSO–GELS algorithm achieves makespan reduction of 29.2% over Simulated Annealing (SA) algorithm for 5000 tasks and 30 resources. A combination of PSO and Pareto optimization has been presented for independent task scheduling in cloud aiming to minimize makespan and cost.

Pacini et al. [20] proposed a virtual machine scheduling algorithm based on IaaS model which aimed to serve multiple users running parameter sweep experiments on private clouds. The performance metrics considered are number of users effectively served which is associated with throughput and the total number of virtual machines allocated which is related to response time. The proposed approach is compared with GA and Random approach. Random approach serves many users, but may not be fair with response time of users as it creates less number of virtual machines. GA serves less number of users and creates more number of virtual machines. The proposed PSO approach serves more users than GA and creates more virtual machines than Random. PSO achieves an excellent balance between number of serviced users and number of created virtual machines with 29.41% gain over GA and 35.29% gain over Random when the number of users is 100 and they connect every 90 s.

Liu and Wang [21] presented an algorithm based on PSO to balance the load between virtual machines in cloud. The algorithm tries to minimize makespan and maximize resource utilization of virtual machines. They modified the basic PSO by introducing a self-adapting inertia weight which is based on particle's fitness value and global best fitness value. A simple mutation mechanism is used in which a random value from solution space is assigned to position if there is an overflow.

Sidhu et al. [22] proposed a load balancing strategy (PSOLR) for heterogeneous computing systems which is applied after scheduling the tasks using discrete PSO technique. The load balancing strategy moves the smallest task from the machine having highest execution span to any other machine which reduces the makespan of the whole schedule. The process is repeated for the remaining N-2 machines and finally the schedule is updated. The iterations are repeated till make-span cannot be reduced further. The experimental results divulge that there is a makespan reduction of 19.6% and 37% over PSO–SPV for homogeneous and heterogeneous environment respectively. Moreover the average resource utilization of PSO-LR is between 18% and 22% and PSO–SPV is 12–31% taking five different kinds of machines and 100 heterogeneous tasks.

2.5 League championship algorithm

Kashan [23] proposed a novel meta-heuristic algorithm termed as League Championship Algorithm (LCA) for global optimization in 2009. It is inspired by the contests of sport teams in a sports association (league). A league schedule is designed every week (iteration) for the teams (individuals) to play in pairs and the result is in the form of win or loss depending upon the playing strength (fitness value) of a team following a meticulous team formation/playing technique (solution). On the basis of prior week knowledge, the team makes changes in the formation (a new solution) for the next week competition and the championship continues till the specified number of seasons (terminating condition). An extensive survey of applications of LCA and its future scope in other application areas has been done. LCA has been used to solve various optimization problems out of which some are traveling salesperson problem, reactive power dispatch problem, job shop scheduling, and optimization of electromagnetic devices, task scheduling in cloud, etc.

Abdulhamid et al. [24] and *Sun et al. [25]* have used this algorithm for solving optimization problems related to cloud scheduling. The authors aimed to minimize makespan of a given set of tasks in Infrastructure as a Service (IaaS) cloud. Their results show that it performs better than First Come First Serve (FCFS), Last Job First (LJF) and Best Effort First (BEF). The algorithm has been implemented in MATLAB. They have proposed a double combinatorial auction based resource allocation mechanism considering the features of cloud resources. They used LCA algorithm to solve winner determination problem of this strategy and aimed to maximize market surplus and overall reputation. It is implemented in SimJava 2.0 toolkit on the Eclipse platform.

2.6 BAT algorithm

Getting inspiration from echolocation behavior of bats, *Yang [26]* introduced BAT algorithm, a novel optimization algorithm in 2010. Bats use echolocation to estimate the distance of their prey. They fly randomly with a velocity, position, frequency, loudness and pulse emission rate to seek for their prey. Procedure LCA When they are hunting for their prey, they can adjust their frequency, loudness and pulse rate of emission based on the distance amid them and the prey. This behavior of bats has been used to formulate BAT algorithm. The change of velocities and positions of bats has a resemblance to PSO algorithm. BAT algorithm can be thought as a hybrid of PSO and the exhaustive local search restricted by loudness and pulse rate. Jacob [93] applied BAT algorithm for resource scheduling in cloud aiming to minimize makespan and concluded that it has high accuracy and efficiency than GA. Kumar et al. [94] proposed an approach for task scheduling in cloud based on the combination of BAT algorithm and Gravitational scheduling algorithm (GSA) considering deadline constraints and trust model. Resources for the tasks are selected on the basis of their trust value. The proposed algorithm is implemented in CloudSim and efficiently reduces makespan and reduces the number of failed tasks in comparison with Random resource selection with GSA. *Raghavan et al. [27]* have used Bat algorithm to solve workflow scheduling problem in cloud aiming to minimize processing cost of the whole workflow. The algorithm performs better in terms of processing cost when compared with Best Resource selection (BRS) algorithm. A hybrid of PSO and Multi-Objective Bat Algorithm is discussed for profit maximization in cloud. PSO is used for local search and global update is done by Bat algorithm as Bat algorithm has high global convergence. M/M/m queuing model is used to manage multi-server system and resources are allocated considering service charge and business cost to maximize profit. Resource provisioning is done according to admission control and profit aware SLA.

2.7 Pareto optimization

Pareto optimization is widely used to solve multi-objective optimization problems having conflicting objectives. Solutions that provide reasonable trade-offs among different objectives are considered. Rather than constructing a single solution, multiple solutions are generated that satisfy Pareto optimality criterion. A solution S is chosen only if no solution is better than S taking into account entire objectives. Suppose if S is worse than some solution S_0 with respect to one objective, S is chosen given that it is better than S_0 with respect to some other objective. Hence every Pareto optimal solution is good with respect to some optimization criterion. The set of all Pareto optimal solutions makes Pareto front/Pareto set. A Pareto-based GA approach has been proposed in [28] for finding best virtual machine instances provided by IaaS provider to fit the client's virtual machine requests in the cloud brokering environment. The approach intended to minimize the response time and the cost of chosen virtual machine instances for client satisfaction and to maximize broker's earnings. Pareto approach is chosen to provide broker as many non-dominated solutions as possible allowing a trade-off between response time and cost. Another hybrid approach of GA and Pareto optimization is introduced in [29] to perform resource allocation optimizing makespan and energy consumed by servers and switches. Implementation is done on an open source called jMetal which provides genetic multi-objective framework. The algorithm is having quadratic time complexity with respect to allocated number of tasks. NSGA II is the evolutionary core of both the algorithms.

3. Findings

Based on the survey, following observations have been made:

(a) Improving quality of solution by combining metaheuristic algorithm with some other algorithm: A metaheuristic algorithm can be improved in terms of quality of the solution or convergence speed by combining it with other population-based metaheuristic algorithm or some local search-based metaheuristic algorithm. One of the advantages of combining two population based metaheuristics is that the shortcomings of one algorithm can be overcome by strengths of other algorithm.

Wen et al. [3] have combined ACO with PSO so that the algorithm should not premature into local optimal solution making inefficient resource scheduling.

Mathiyalagan et al. [30] proposed a hybridization technique using ACO and Intelligent Water drops (IWD) algorithm, a recent population based metaheuristic to improve performance in terms of execution speed and quality of solution. *Raju et al. [31]* combined ACO with Cuckoo Search to get the advantages of both the algorithms.

Local search-based algorithms can be used to further improve the solution of population-based metaheuristic algorithms. The best regions in search space of problem are identified using population based metaheuristic whereas local search techniques help in finding optimum solutions in those best regions. In this context, has incorporated local search strategy at the end of each iteration of ACO to improve each obtained solution, applied Simulated Annealing (a local search based metaheuristic) after selection, crossover and mutation in each iteration of Genetic algorithm used Variable Neighborhood Search (VNS) on the solution given by PSO, and GA used hill climbing with PSO. A hybrid of PSO and Tabu Search (TS) is used to enhance resource utilization and reduce energy consumption.

(b) Improving quality of solution by initial population generation:

Quality of solutions of population-based metaheuristic algorithms such as GA and PSO can be improved by generating initial population using local search techniques. It uses Minimum Execution Time (MET) and Min–min heuristic, used Longest Job to Fastest Processor and Smallest Job to Fastest Processor heuristics to create initial population of GA [32-35]. They created initial particles of PSO based on Shortest Job to Fastest Processor (SJFP) Algorithm, whereas Greedy Randomized Adaptive Search Procedure (GRASP) for this purpose. The elite solutions that are selected best solutions from generations can also be used to generate initial populations for upcoming generations. These elites if enhanced, before becoming part of next generation can attain better performance than those achieved by original elites.[36]

(c) Improving quality of solution by modifying the transition operator: Researchers have focused on modifying the transition operators used in metaheuristic algorithms. In case of ACO, various strategies have been proposed for pheromone updation. The updation of pheromone strategy decides the selection of ants for updation process and what they need to do once they are selected. This greatly affects the search strategy of ACO.

(d) Energy conservation: For Energy conservation, VM placement optimization, VM consolidation and DVFS techniques are popularly used. The main drawback of using DVFS technique is that the frequency and voltage can only be adjusted to limited values. The techniques also vary from each other based on whether they are considering single resource (i.e. CPU utilization, as CPU utilization consumes maximum power as compared to other resources) or multiple resources (RAM, disk and network bandwidth).

4. Conclusion

The study provides a comprehensive evaluation of the use of metaheuristic approaches for scheduling in cloud and grid systems. The majority of research is focused on increasing the convergence speed and quality of the solution because metaheuristic techniques are frequently slower than deterministic algorithms and the generated solutions may not be ideal. Modifying the transition operator, preparing the input population, or using a hybrid method with metaheuristic techniques have all been used to address these concerns. Additionally, many scheduling methods have concentrated on various optimisation criteria. Most writers in the reviewed literature concentrated on lowering makespan and execution costs, while others placed importance on reaction time, throughput, flow time, and average resource utilisation. When comparing algorithms built using different metaheuristic techniques, the major factors to consider are the method for enhancing metaheuristics, the optimisation criteria, the types of tasks involved, and the environment in which the algorithm is employed. As data centres have grown to be energy hogs and a significant source of CO emissions, new research efforts have been made in the direction of energy-aware scheduling. The difficulty lies in lowering data centre energy consumption without compromising performance or going against SLA requirements. The report also discusses a number of unresolved concerns that can be the subject of future studies.

5. Future Research

Future research should concentrate on finding solutions to the problems that academics are now facing and on coming up with new ideas to expand the cloud horizon. The following efforts can be done in the future to address the limitations of current algorithms;

- Creation of resource-constrained efficient protocols and algorithms that don't compromise the high level of security required to secure sensitive data.
- Machine learning techniques could be utilised as a service to lower the likelihood of failure in reliability. It is difficult to estimate the upcoming workload in a cloud computing environment, hence more powerful machine learning methodologies and prediction techniques must be developed.
- A business model is cloud. The primary user concerns, such as security, dependability, and SLA maintenance, should be focused on when executing applications so that a productive and secure business cloud framework can be created.
- Because green computing is expected to become a pressing issue for the cloud in the near future, extensive study is needed to schedule resources depending on energy. There are some fundamental controls that must be organised, such as task migration, VM migration, monitoring memory usage, setting a threshold value, etc.

References

[1] Chen W, Shi Y, Zhang J. An ant colony optimization algorithm for the time-varying workflow scheduling problem in grids. IEEE, Congr Evol Comput 2009:875–80.

[2] Hu Y, Xing L, Zhang W, Xiao W, Tang D. A knowledge-based ant colony optimization for a grid workflow scheduling problem. In: Adv Swarm Intell Notes Comput Sci. Springer; 2010. p. 241–8. http://dx.doi.org/10.1007/978-3-642-38703-6.

[3] Wen X, Huang M, Shi J. Study on resources scheduling based on ACO algorithm and PSO algorithm in cloud computing. In: Proc– 11th int symp distrib comput appl to business eng sci; 2012. p.219–22. http://dx.doi.org/10.1109/DCABES.2012.63.

[4] Pacini E, Mateos C, Garcia C. Balancing throughput and response time in online scientific clouds via ant colony optimization. Adv Eng Software 2015;84:31–47 [Elsevier].

[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. http://dx.doi. org/10.1201/9781420038347.

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

[7] Ge Y, Wei G. GA-based task scheduler for the cloud computing systems. In: Proc int conf web inf syst min, vol. 2; 2010. p. 181–6.http://dx.doi.org/10.1109/WISM.2010.87.

[8] Zhao C, Zhang S, Liu Q, Xie J, Hu J. Independent tasks scheduling based on genetic algorithm in cloud computing. In: 5th int conf wirel commun netw mob comput; 2009. p. 1–4. http://dx.doi.org/10.1109/WICOM.2009.5301850.
[9] Zheng Z, Wang R, Zhong H, Zhang X. An approach for cloud resource scheduling based on parallel genetic algorithm. In: 3rd IEEE int conf comput res dev. IEEE; 2011. p. 444–7. http://dx. doi.org/10.1109/ICCRD.2011.5764170.
[10] Ghorbannia Delavar A, Aryan Y. HSGA: a hybrid heuristic algorithm for workflow scheduling in cloud systems. Cluster Comput 2014;17:129–37. http://dx.doi.org/10.1007/s10586-0130275-6.
[11] Wang T, Liu Z, Chen Y, Xu Y, Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. In: IEEE 12th int conf dependable auton secur comput; 2014. p. 146–52. http://dx.doi.org/10.1109/DASC.2014.35.
[12] Gu J, Hu J, Zhao T, Sun G. A new resource scheduling strategy based on genetic algorithm in cloud computing environment. Comput 2012;7:42–52. http://dx.doi.org/10.4304/jcp.7.1.42-52.
[13] Zhu K, Song H, Liu L, Gao J, Cheng G. Hybrid genetic algorithm for cloud computing applications. In: IEEE Asia Pacific serv comput conf; 2011. p. 182–7. http://dx.doi.org/10.1109/APSCC.2011.66.
[14] Wang T, Liu Z, Chen Y, Xu Y, Dai X. Load balancing task scheduling based on genetic algorithm in cloud computing. In: IEEE 12th int conf dependable auton secur comput; 2014. p. 146–52. http://dx.doi.org/10.1109/DASC.2014.35.
[15] Shen G, Zhang YQ. A shadow price guided genetic algorithm for energy aware task scheduling on cloud computers. Adv Swarm Intell Notes Comput Sci, vol. 6728. Springer; 2011. p. 522–9.http://dx.doi.org/10.1007/978-3-642-21515-5_62.
[16] Liu H, Abraham A, Hassanien AE. Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm. Futur Gener Comput Syst 2010;26:1336–43. http://dx.doi.org/10.1016/j.future.2009.05.022.
[17] Wu Z, Ni Z, Gu L, Liu X. A revised discrete particle swarm optimization for cloud workflow scheduling. In: Proc – 2010 int conf comput intell secur cis. IEEE; 2010. p. 184–8. http://dx.doi.org/10.1109/CIS.2010.46.
[18] Zhang L, Chen Y, Sun R. A task scheduling algorithm based on PSO for grid computing. Int J Comput Intell Res 2008;4:37–43. http://dx.doi.org/10.1109/ISDA.2006.253921.
[19] Pooranian Z, Shojafar M, Abawajy JH, Abraham A. An efficient meta-heuristic algorithm for grid computing. J Comb Optim 2013:1–22. http://dx.doi.org/10.1007/s10878-013-9644-6.
[20] Pacini E, Mateos C, Garino CG. Dynamic scheduling based on particle swarm optimization for cloud-based scientific experiments. CLEI Electron J 2014;14:1–12.
[21] Liu Z, Wang X. A PSO-based algorithm for load balancing in virtual machines of cloud computing environment. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2012;7331 LNCS:142–7. http://dx.doi.org/ 10.1007/978-3-642-30976-2_17.
[22] Sidhu MS, Thulasiraman P, Thulasiram RK. A load-rebalance PSO heuristic for task matching in heterogeneous computing systems. In: Proc IEEE symp swarm intell SIS 2013, IEEE symp ser comput intell SSCI 2013; 2013. p. 180–7. http://dx.doi.org/10. 1109/SIS.2013.6615176.
[23] Abdulhamid SM, Latiff MSA, Madni SHH, Oluwafemi O. A survey of League Championship Algorithm: prospects and challenges. Indian J Sci Technol 2015;8:101–10.
[24] Abdulhamid SM, Latiff MA, Idris I. Tasks scheduling technique using League Championship Algorithm for makespan minimization in IaaS cloud. ARPN J Eng Appl Sci 2014;9:2528–33.
[25] Sun J, Wang X, Li K, Wu C, Huang M, Wang X. An auction and League Championship Algorithm based resource allocation mechanism for distributed cloud. Lect Notes Comput Sci (Including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) 2013;8299:334–46. http://dx.doi.org/10.1007/ 978-3-642-45293-2_25.
[26] Yang XS. A new metaheuristic bat-inspired algorithm. Nat Inspired Coop Strateg Optim Comput Intell 2010;284:65–74. http://dx.doi.org/10.1007/978-3-642-12538-6_6.
[27] Raghavan S, Marimuthu, C, Sarwesh, P, & Chandrasekaran K. Bat algorithm for scheduling workflow applications in cloud. Int Conf Electron Des Comput Networks Autom Verif (EDCAV). IEEE; 2015. p. 139–44.
[28] Kessaci Y, Melab N, Talbi EG. A pareto-based genetic algorithm for optimized assignment of VM requests on a cloud brokering environment. IEEE congr evol comput; 2013. p. 2496–503. http://dx.doi.org/10.1109/CEC.2013.6557869.
[29] Portaluri G, Giordano S. A power efficient genetic algorithm for resource allocation in cloud computing data centers. In: 3rd int conf cloud netw. IEEE; 2014. p. 58–63.
[30] Mathiyalagan P, Sivanandam SN, Saranya KS. Hybridization of modified ant colony optimization and intelligent water drops algorithm for job scheduling in computational grid. ICTACT J SOFT Comput 2013;4:651–5.
[31] Raju R, Babukarthik RG, Chandramohan D, Dhavachelvan P, Vengattaraman T. Minimizing the makespan using Hybrid algorithm for cloud computing. In: Proc 3rd IEEE int adv comput conf; 2013. p. 957–62. http://dx.doi.org/10.1109/IAdCC. 2013.6514356.
[32] V Naiyer, J Sheetlani, HP Singh 2020, 'Software Quality Prediction Using Machine Learning Application', Smart Intelligent Computing and Applications: Proceedings of the Third International Conference on Smart Computing and Informatics, Volume 2 pp. 319-32.
[33] R Kumar, HP Singh, GA Kumar 2020, 'Design and Development of Performance Evaluation Model for Bio-Informatics Data Using Hadoop', Solid State Technology, volume 63 issue 6, pp. 23284-23296.
[34] BV Laxmi 2021, 'A Review of Dynamic Resource Allocation Framework for Large Amount of Cloud Enterprises, Turkish Journal of Computer and

Mathematics Education, volume 12 issue 2, pp. 1280-1284.

[35] Sheetlani et al., "A Proposed Framework Of Dimensionality Reduction Techniques To Boost Credit Card Fraud Classification", IJFANS International Journal of Food and Nutritional Sciences, Volume 11, S Iss 3, Dec 2022.

[36] Singh et al., "Cloud computing security issues, challenges and solutions", EasyChair, 2020, issue 2533.