



# Trust Based Resource Selection in Cloud Computing Using Hybrid Algorithm

**V Suresh Kumar**

*Research Scholar,  
M S University Tirunelveli, Tamilnadu, India  
sureshkumarv.clo@gmail.com*

**Aramudhan**

*Department of Information Technology  
Perunthalaivar Kamarajar Engineering College  
Tamilnadu, India*

**Abstract-** Cloud computing is experiencing rapid advancement in academia and industry. This technology offers distributed, virtualized and elastic resources as utilities for end users and can support full recognition of “computing as a utility” in the future. Scheduling distributes resources among parties which simultaneously and asynchronously seek it. Scheduling algorithms are meant for scheduling and they reduce resource starvation ensuring fairness among those using the resources. Most Task-scheduling cloud computing procedures consider task, resource requirements for CPU and memory, and not bandwidth. This study suggests optimizing scheduling with BAT-Harmony search hybrid algorithm.

Keywords- Cloud Computing, Task Scheduling, BAT algorithm, Harmony search

## I. INTRODUCTION

Cloud computing uses central remote servers and internet for data and application maintenance. It permits consumers to use applications sans installation and access personal files on a computer having internet access. This ensures more efficient computing through centralizing memory, storage, processing and bandwidth [1]. Cloud computing is “a type of parallel and distributed system in a collection of interconnected and virtualized computers dynamically provisioned and accessed as one or more unified computing resources established on Service Level Agreement (SLA), via negotiation between service provider and consumer”.

Due to benefits like elastic and scalable, on-demand resources, Cloud service users explore the ability of scalable platforms for efficient and cost-effective execution of applications like e-commerce, High Performance Computing (HPC), social network and web applications. To successful Cloud resources use, applications should adapt to the new environment and new scheduling solutions should be developed to ensure good performance. So also, Cloud providers should determine proper configurations and scheduling policies for proper use of computer networks and storage resources like different applications that are executed both concurrently and in isolation [2].

To ensure that cloud services are effective as provider infrastructure, task scheduling algorithm is a major requirement. They are responsible for map jobs submitted a cloud environment to resources so that total response time in makespan, is reduced [3]. Task scheduling is allocating one or more time intervals to one or more resources. Scheduling is a problem of scheduling submitted tasks set for different users of computing resources to reduce a specific job completion time or system makespan. Many parameters are factors for scheduling problems like system throughput, load balancing, service cost, service reliability, and system use.

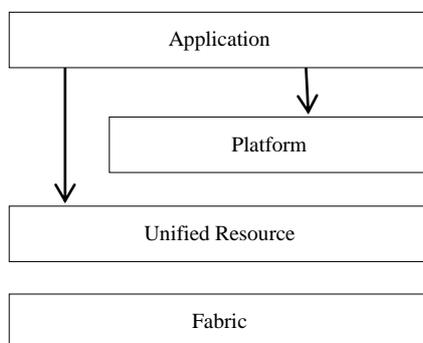


Figure 1. Cloud Architecture

Scheduling is a set of policies controlling the order of work a computer system has to perform. Various scheduling algorithms exist in distributed computing, and job scheduling is one. The advantage of the job scheduling algorithm is achieving high performance computing and best system throughput. Scheduling

manages CPU, memory availability. A good scheduling policy ensures maximum resource use [4]. Job grouping based scheduling is a dynamic scheduling strategy, maximizing resources use, processing capabilities and reducing overhead time and cost to execute jobs.

Job scheduling considers: (i) processing requirements for a job, (ii) jobs grouping mechanism, called job grouping, according to resources processing capabilities, and (iii) job grouping transmission to correct resource [5]. The scheduler orders jobs so that the balance between improving quality of services and simultaneously maintaining job efficiency and fairness is maintained. So, evaluating scheduling algorithm's performance is crucial to realizing large-scale distributed systems. Despite various scheduling algorithms for cloud environments, no comprehensive performance study providing a unified platform for comparing such algorithms exists [6].

As Cloud computing Systems users increased, tasks to be scheduled also increases proportionally. So, better algorithms to schedule tasks on systems are required. Algorithms to schedule tasks are service oriented and differ according to the environments [7]. Cloud computing Task Scheduling algorithms minimize tasks makespan with minimum resources. Cloud computing, uses low-power hosts to ensure high usability. Cloud computing is a class of systems and applications using distributed resources to function in a decentralized way.

Available resources must be used efficiently without affecting service parameters of cloud. Cloud scheduling process service parameters are generalized in 3 stages namely [8],

1. *Resource discovering and filtering* - Datacenter Broker notices resources in network and gets their status information.
2. *Resource selection* - Target resource selection is based on task and resource parameters. This is a deciding stage
3. *Task submissions* - Tasks are submitted to selected resources

Task scheduling is a famous combinatorial optimization problem. The aim is determining a proper sequence during task execution while obeying some (transaction logic) constraints. As an optimization problem, getting an optimal packing (allocation) scheme needs huge computation time. After a resource *calculation* algorithm determines mapping scenarios to satisfy QoS requirements, an actual resource allocation algorithm maps task demands to available resources [9].

The target of scheduling is maximizing resource use and lowering tasks process time. An efficient job scheduling strategy must yield less response time [10] to ensure that jobs execution takes place in a stipulated time. Simultaneously there is an occurrence of in time resource reallocation. Due to this, jobs take place and more jobs are submitted to cloud by clients which results in accelerating cloud system's business performance. Many scheduling algorithms are available and the most popular are discussed below.

*FCFS* is First Come First Serve algorithm where the first data reaching queue first is executed first. The *round robin algorithm* has an advantage over FCFS algorithm. It allocates a process or task with a time slot and after this, the job is changed and the next job comes to execution. *Priority scheduling* algorithm discards FCFS and round robin algorithm's disadvantages. In this algorithm, each job's priority is decided based on tasks properties [11]. *Hybrid algorithm* uses FCFS and a priority algorithm where a job queue has only one algorithm to be done, when it goes for FCFS algorithm, but when the next algorithm reaches the queue, it checks jobs priority and reschedule tasks accordingly. A hybrid approach is always better than a single approach.

Many-task workflow scheduling in a distributed computing platform is an NP-hard problem which is even more complex and challenging when virtualized clusters execute many tasks on a cloud computing platform. The difficulty is in satisfying multiple objectives that may be conflicting in nature. It is difficult to minimize many tasks makespan, when reducing resource cost and preserving fault tolerance and/or Quality of Service (QoS) simultaneously. Such requirements and goals are hard to optimize due to unknown runtime conditions, like resource availability and random workload distributions [12].

Usually, standard BA algorithm exploits search space, but sometime is trapped in local optima and so cannot perform global search well. For BA, search depends on random walks and hence fast convergence is not guaranteed. This study proposes a hybrid algorithm for BAT-Harmony search for scheduling optimum in cloud computing. The rest of the study is as follows: Section 2 reviews related works in literature. Section 3 explains the methods used in the proposed work; Section 4 discusses experiments and results and Section 5 concludes the work.

## II. RELATED WORK

A service flow scheduling with various QoS requirements in cloud computing was proposed by Liu et al., [13], adopting an Ant Colony Optimization (ACO) algorithm to improve service flow scheduling. In the new model, default rate described radio of cloud service provider breaking SLA, and introduced SLA module to monitor cloud services running state.

The relation between infrastructure components and power consumption of a cloud computing environment was studied by Luo et al., [14], who discussed task types matching and component power adjustment methods. A Cloud Computing resource scheduling algorithm built on energy efficient optimization methods was presented. Experiments proved that jobs that used hardware environment had energy consumption reduced by the new algorithm.

An ACO based job scheduling algorithm, adapted to dynamic cloud computing characteristics that integrated specific ACO advantages in NP-hard problems was proposed by Song et al., [15]. It aimed to reduce job completion time based on pheromone. Experiments showed that the ACO algorithm is promising for job scheduling in cloud computing environments.

System service cost and task scheduling algorithms of optimizing resource were studied by Wng et al., [16]. Besides, it also proposed a mathematical model of Dynamic Service-quality Cost (DSC) which in real-time synthetically measures service cost and node's service performance in cloud computing. Further, it also presented a complexity-aware optimization task scheduling algorithm. Finally, experiments proved the proposed algorithm's reasonability and effectiveness. It achieved not only reduced service cost, but also increased service quality by reasonably assigning tasks to nodes for calculation in cloud computing. And results were consistent with theoretical analysis.

A PSO based heuristic to schedule applications to cloud resources that considered the computation cost and data transmission cost was presented by Pandey et al., [17]. A workflow application was experimented on varying computation and communication cost. Cost savings when using PSO was compared with Best Resource Selection (BRS) algorithm. Results revealed that PSO achieved: (a) three times more cost savings compared to BRS, and (b) also resulted in better workload distribution.

The use of working scheduling models to improve resource allocation in cloud computing was suggested by Chen and Tseng [18]. However, cloud computing and resource distribution are more responsive with applications needing to be more time and resources optimizing oriented. Authors analog users in accordance with demand work category parameters, and scheduling parameters to access Internet usage or take single time to work the parameter, by queuing theory for scheduling models like demand, scheduling parameters to model, generating reference data, through data analysis to compare various scheduling characteristics. Cloud computing systems future is a frame of reference to build own cloud computing models.

A new method to solve problems by applying stochastic integer programming for ideal resource scheduling in cloud computing was proposed by Li and Guo [19]. Applying Grobner base theory to solve stochastic integer programming problem and experimental results of implementation are presented.

Atleast Languages First Min-Min (LLFMM), algorithm based scheduling algorithm, suitable to multilingual information resources to improve information resources, scheduling performance in cloud computing was proposed by Han and Luo [20]. Concurrent Scheduling Marking Graph (CSMG) of Colored Dynamic Timed Petri nets (CDTdPN) was defined and algorithm for constructing CSMG of CDTdPN was given. CSMG represented the concurrent relation of transitions in Petri nets and reduced graph size to be constructed to some extent. Finally, example and simulation revealed that the new scheduling algorithm reduced makespan and that the new models and analyzing techniques were convenient to analyze scheduling system performance.

A QoS guided task scheduling model, composed of scheduling strategies and QoS guided scheduling Sufferage-min heuristic algorithm was presented by Han et al., [21]. The experiment revealed that make span value was successfully shortened.

A Multi Queue Scheduling (MQS) algorithm to reduce the cost of the reservation and on-demand plans using global scheduler was proposed by Karthick et al., [22]. Scheduling is an important and complex part in cloud computing. The new methodology depicted the clustering concept for jobs based on burst time. The new method overcame problems and reduced starvation in the process. The new MQS method gave more importance to job selection dynamically to achieve an optimum cloud scheduling problem, and hence it used the unused free space economically.

A Period ACO (PACO) based scheduling algorithm, to solve task scheduling problems in cloud computing was proposed by Sun et al., [23]. PACO used ACO algorithm in cloud computing, with first proposed scheduling period strategy and improvement of pheromone intensity updated strategy. Experiment showed that PACO ensured good performance in makespan and load balance of the total cloud cluster.

A new framework built combining computing resources on Cloud computing and computing components in local systems was presented by Man and Huh [24]. The framework's main component is cost with finish time-based scheduling algorithm that balanced application schedule performance and mandatory cost for use of Cloud resources. Experiments and comparison with the other scheduling approaches proved benefits of the new algorithm.

An architecture based on collaboration between thin-thick clients and clouds that provided effective task scheduling which boosted processing time in mobile cloud platform was proposed by Hung et al., [25] while considering network bandwidth and cost for cloud service usage. Experiments showed improved efficiency of task scheduling in ensuring desired processing time that corresponded to customer's money.

An optimization model for task scheduling for reducing energy consumption in cloud-computing data centers was presented by Liu et al., [26]. The new approach was formulated as an integer programming problem to lower cloud-computing data center energy consumption by scheduling tasks to a minimum servers while keeping task response time constraints. Additionally, the most efficient server first task-scheduling scheme to minimize energy expenditure as a practical scheduling scheme was formulated. Simulation show that new task-scheduling scheme reduced server energy consumption on average over 70 times compared to energy consumed under a (not-optimized) random-based task-scheduling scheme. Energy savings were achieved by reducing allocated servers.

### III. METHODOLOGY

This study proposed a hybrid method using BAT algorithm and Harmony search algorithm for scheduling optimization in cloud computing.

#### A. BAT algorithm

Xin-She Yang in 2010 [27] developed BAT algorithm which exploits the bats echolocation. Bats use sonar echoes to detect/avoid obstacles. Sound pulses are transformed to a frequency that reflects from obstacles. Bats use time delay from emission to reflection and use it to navigate. They emit short loud, sound impulses and the pulse rate is defined as 10 to 20 times /sec. After hitting and reflecting, bats transform own pulse to advantageous information to measure how far away the prey is. Bats use wavelengths that vary from in range [0.7, 17] mm or inbound frequencies [20,500] kHz.

BAT algorithm is based on idealizing bats' echolocation characteristics which follow approximate or idealized rules [28]:

- (1) Bats use echolocation to sense distance, and "know" surroundings in some magical way.
- (2) Bats fly randomly with velocity  $v_i$  and a fixed frequency  $f_{\min}$  at position  $x_i$ , varying wavelength  $\lambda$ , and loudness  $A_0$  to hunt. They spontaneously accommodate wavelength (or frequency) of emitted pulses and adjust pulse rate emission  $r \in [0, 1]$ , depending on target's proximity.
- (3) Though loudness changes in different ways, it is supposed that loudness varies from a minimum constant (positive)  $A_{\min}$  to a large  $A_0$ .

#### B. Harmony search

Harmony Search (HS) is a music-based meta-heuristic optimization algorithm inspired by observation that music aims to search for a perfect state of harmony. The harmony in music is analogous to finding optimality in optimization. Optimization, search process is compared to a jazz musician's improvisation process. On one hand, perfectly pleasing harmony is determined by audio aesthetic standards. A musician intends to produce music with perfect harmony. Alternatively, an optimal solution to an optimization problem is the best solution to a problem under given objectives and limited by constraints. Both processes produce best or optimum [29].

HS algorithm initializes Harmony Memory (HM) with random generated solutions. The solutions stored in HM are defined by Harmony Memory Size (HMS). Then a new solution is created as follows iteratively. Every decision variable is generated on memory consideration and a possible additional modification, or through random selection. Parameters used in the generation of a new solution are called Harmony Memory Considering Rate (HMCR) and Pitch Adjust Rate (PAR). Each decision variable is set to value of the corresponding variable of one solution in HM with a HMCR probability. Further modification of this value is done with a PAR probability [30]. Otherwise (with a probability of  $1 - \text{HMCR}$ ), decision variable is set to random value. After a new solution is generated, it is evaluated and compared to worst HM solution. If its value is better than that of worst solution, it replaces a worst solution in HM. This is repeated, till a termination criterion is fulfilled. The hybrid HS/BA meta-heuristic algorithm is as follows [31]:

#### Begin

- Step 1: Initialization.** Set the generation counter  $t = 1$ ;  
 initialize the population of NP bats  $P$  randomly and  
 each bat corresponding to a potential solution to the given problem;  
 define loudness  $A$ ; set frequency  $Q$ , the initial velocities  $V$ , and pulse rate  $r$ ;  
 set the harmony memory consideration rate HMCR,

the pitch adjustment rate PAR and bandwidth bw;

set maximum of elite individuals retained KEEP.

**Step 2:** Evaluate the quality  $f$  for each bat in  $P$  determined by the objective function ( $x$ ).

**Step 3: While** the termination criterion is not satisfied or  $t < \text{MaxGeneration}$  **do**

Sort the population of bats  $P$  from best to worst by order of quality  $f$  for each bat.

Store the KEEP best bats as KEEPBAT.

**for**  $i = 1:\text{NP}$  (all bats) **do**

$$v_i^t = v_i^{t-1} + (v_i^t - x_{*}) \times Q$$

$$x_i^t = x_i^{t-1} + v_i^t$$

**if**(rand >  $r$ ) **then**

$$x_u^t = x_{*} \alpha \varepsilon^t$$

**end if**

**for**  $j = 1:D$  (all elements) **do** //Mutate

**if**(rand < HMCR) **then**

$$r_1 = \lceil \text{NP} * \text{rand} \rceil$$

$$x_v(j) = x_{r_1}(j) \quad \text{where } r_1 \in (1, 2, \dots, \text{HMS})$$

**if**(rand < PAR) **then**

$$x_v(j) = x_v(j) + bw \times (2 \times \text{rand} - 1)$$

**endif**

**else**

$$x_v(j) = x_{\min,j} + \text{rand} \times (x_{\max,j} - x_{\min-j})$$

**endif**

**endfor**  $j$

Evaluate the fitness for the offsprings  $x_u^t, x_i^t, x_v^t$

Select the offspring  $x_k^t$  with the best fitness among the offsprings  $x_u^t, x_i^t, x_v^t$

**if**(rand <  $A$ ) **then**

$$x_{r_1}^t = x_k^t$$

**end if**

Replace the KEEP worst bats with the KEEP best bats KEEPBAT stored.

**end for**  $i$

$t = t + 1$ ;

**Step 4: end while**

**Step 5:** Post-processing the results and visualization;

**End**

Generally Trust establishes/maintains relationships between 2 entities for long. Applying trust models to scheduling decrease failure ratio and reassigns cloud environments. Combining communication, trust and data trust locates a component/resource/service's overall trust when scheduling. Data's trust decides resources list to calculate trust and threshold levels to separate trustful and untrusted nodes. Communication trust is calculated

on client's bandwidth availability and resource centers. Bayesian model is used for data fusion and communication trusts. Reputation ratings are calculated by probability density functions based beta reputation given by [32],

$$f(p|\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{1}$$

Where  $\alpha$  represents jobs completed and  $\beta$  represents unsuccessful jobs.  $R_{ij}$  is reputation for resource  $n_i$  observed from neighborhood resources  $n_j$ .

$$R_{ij} = \text{Beta}(\alpha_{ij} + 1, \beta_{ij} + 1) \tag{2}$$

Trust value is then calculated using expected reputation value.

$$T_{ij} = E(R_{ij}) = E\{\text{Beta}(\alpha_{ij} + 1, \beta_{ij} + 1)\} = \frac{(\alpha_{ij} + 1)}{(\alpha_{ij} + \beta_{ij} + 2)} \tag{3}$$

#### IV. EXPERIMENTAL RESULTS

The experiments are conducted with BAT and HS algorithms separately and with the hybrid algorithm. The proposed scheduling mechanism was implemented.

TABLE I. TIME REQUIRED TO COMPLETE TASK

Time required to complete all task	Time (second)
Random resource selection with BAT scheduling	391.14
Trust based resource selection with BAT scheduling	371.26
Random resource selection with Harmony Search scheduling	384.02
Trust based resource selection with Harmony Search scheduling	378.11
Random resource selection with BAT-HS scheduling	366.71
Trust based resource selection with BAT-HS scheduling	352.42

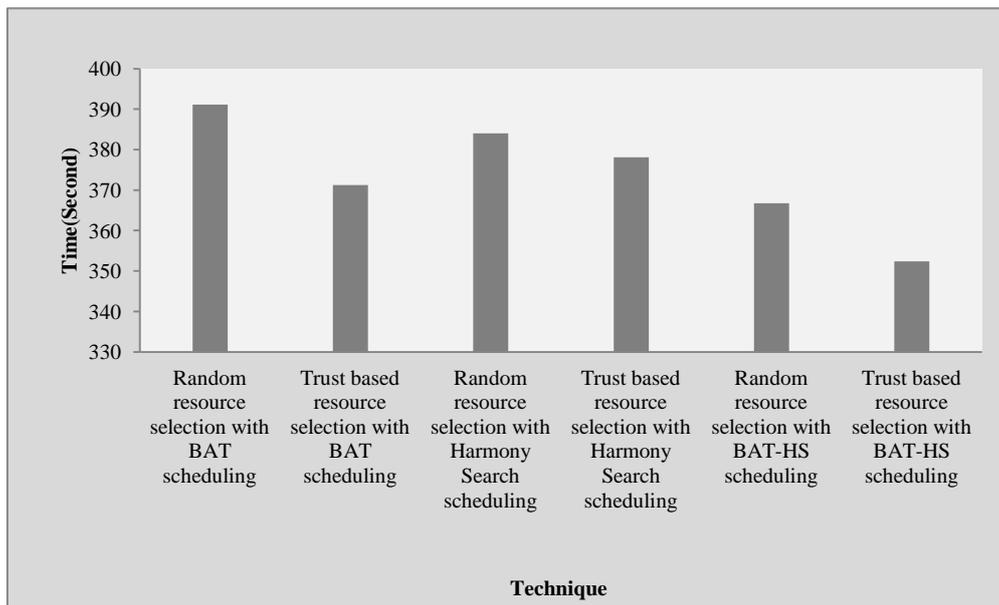


Figure 2. Time required to complete task

Trust based resource selection with BAT-HS scheduling decreases the time essential to complete all tasks by 3.9742% when compared with Random resource selection with BAT-HS scheduling, by 8.5818% when compared with Random resource selection with HS scheduling and by 10.4148% when compared with Random resource selection with BAT scheduling.

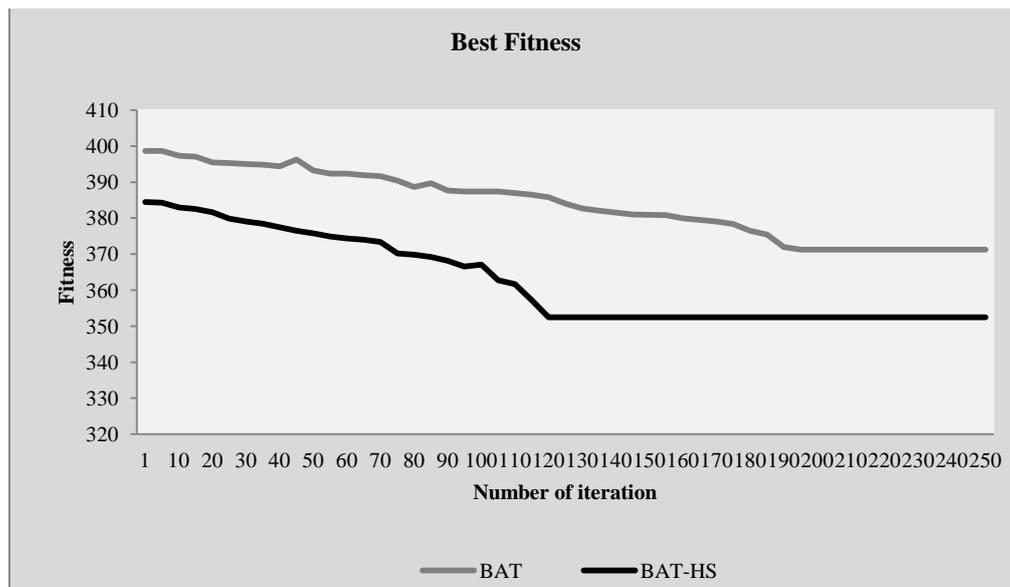


Figure 3. Best Fitness achieved

## V. CONCLUSION

Cloud computing is a paradigm sharing computing and storage infrastructure over scalable resources network. In today's world, data is scattered across different data centers and applications are stored in remote servers. Cloud technology brings scattered data and remote applications to user's laptop virtually. The idea is to make computing and storage infrastructure available to cloud users regardless of time and location. Trust has an important role in commercial cloud environments and trust management is an integral part of cloud technology's commercial aspects. This study proposed a trust based resource selection scheme using BAT-HS for scheduling problem in Cloud computing. The new Trust based resource selection with BAT-HS scheduling decreased time to complete tasks by 10.4148% compared to Random resource selection having BAT scheduling.

## REFERENCES

- [1] Bhatt, K., & Bundele, M. (2013). Review Paper on PSO in workflow scheduling and Cloud Model enhancing Search mechanism in Cloud Computing. *IJNET-International Journal of innovations in engineering and technology*, 2(3).
- [2] Garg, S. K., & Buyya, R. (2011). CloudSim Estimation of a Simple Particle Swarm Algorithm. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(8), (pp. 1279-1287).
- [3] Elzeki, O. M., Reshad, M. Z., & Elsoud, M. A. (2012). Improved Max-Min Algorithm in Cloud Computing. *International Journal of Computer Applications*, 50(12), 22-27.
- [4] Agarwal, D., & Jain, S. (2014). Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment. *arXiv preprint arXiv:1404.2076*.
- [5] Chawla, Y., & Bhonsle, M. Dynamically optimized cost based task scheduling in Cloud Computing.
- [6] Azawi Mohialdeen, I. (2013). Comparative study of scheduling algorithms in cloud computing environment. *Journal of Computer Science*, 9(2).
- [7] Vijayalakshmi A. Lepakshi & Prashanth C S R (2013). A Study on Task Scheduling Algorithms in Cloud Computing. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(11), 119-125.
- [8] Jangra, A., & Saini, T. (2013). Scheduling optimization in cloud computing. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, 3, 62-65.
- [9] Heger, D. A. (2010). Optimized Resource Allocation & Task Scheduling Challenges in Cloud Computing Environments.
- [10] Vijayalakshmi, M., & Muthusamy, K. An Efficient Study of Job Scheduling Algorithms with ACO in Cloud Computing Environment.
- [11] Princy Bathla., Sahil Vashist., Rajwinder Singh., & Gagandeep Singh., (2014). A Sophisticated Review of the Job Scheduling methods on Cloud Network. *International Journal of Latest Scientific Research and Technology (IJLSRT)*.
- [12] Zhang, F., Cao, J., Li, K., Khan, S. U., & Hwang, K. (2014). Multi-objective scheduling of many tasks in cloud platforms. *Future Generation Computer Systems*, 37, 309-320.
- [13] Liu, H., Xu, D., & Miao, H. (2011). Ant colony optimization based service flow scheduling with various QoS requirements in cloud computing. In *Software and Network Engineering (SSNE), 2011 First ACIS International Symposium on* (pp. 53-58).
- [14] Luo, L., Wu, W., Di, D., Zhang, F., Yan, Y., & Mao, Y. (2012). A resource scheduling algorithm of cloud computing based on energy efficient optimization methods. In *Green Computing Conference (IGCC), 2012 International* (pp. 1-6).
- [15] Song, X., Gao, L., & Wang, J. (2011). Job scheduling based on ant colony optimization in cloud computing. In *Computer Science and Service System (CSSS), 2011 International Conference on* (pp. 3309-3312).

- [16] Wang, N., Yang, Y., Meng, K., Chen, Y., & Ding, H. (2013). A task scheduling algorithm based on qos and complexity-aware optimization in cloud computing. In *Information and Communications Technology 2013, National Doctoral Academic Forum* on (pp. 1-8)..
- [17] Pandey, S., Wu, L., Guru, S. M., & Buyya, R. (2010). 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).
- [18] Chen, C. Y., & Tseng, H. Y. (2012). An Exploration of the Optimization of Executive Scheduling in the Cloud Computing. In *Advanced Information Networking and Applications Workshops (WAINA), 2012 26th International Conference* on (pp. 1316-1319).
- [19] Li, Q., & Guo, Y. (2010). Optimization of Resource Scheduling in Cloud Computing. In *SYNASC* (pp. 315-320).
- [20] Han, Y., & Luo, X. (2013). An Effective Algorithm and Modeling for Information Resources Scheduling in Cloud Computing. In *Advanced Cloud and Big Data (CBD), 2013 International Conference* on (pp. 14-19). IEEE.
- [21] Han, H., Deyui, Q., Zheng, W., & Bin, F. (2013). A Qos Guided task Scheduling Model in cloud computing environment. In *Emerging Intelligent Data and Web Technologies (EIDWT), 2013 Fourth International Conference* on (pp. 72-76).
- [22] Karthick, A. V., Ramaraj, E., & Subramanian, R. G. (2014). An Efficient Multi Queue Job Scheduling for Cloud Computing. In *Computing and Communication Technologies (WCCCT), 2014 World Congress* on (pp. 164-166).
- [23] Sun, W., Zhang, N., Wang, H., Yin, W., & Qiu, T. (2013). PACO: A Period ACO Based Scheduling Algorithm in Cloud Computing. In *Cloud Computing and Big Data (CloudCom-Asia), 2013 International Conference* on (pp. 482-486).
- [24] Man, N. D., & Huh, E. N. (2013). Cost and efficiency-based scheduling on a general framework combining between cloud computing and local thick clients. In *Computing, Management and Telecommunications (ComManTel), 2013 International Conference* on (pp. 258-263).
- [25] Hung, P. P., Bui, T. A., & Huh, E. N. (2013). A Thin-Thick Client Collaboration for Optimizing Task Scheduling in Mobile Cloud Computing. In *IT Convergence and Security (ICITCS), 2013 International Conference* on (pp. 1-4).
- [26] Liu, N., Dong, Z., & Rojas-Cessa, R. (2013). Task Scheduling and Server Provisioning for Energy-Efficient Cloud-Computing Data Centers. In *Distributed Computing Systems Workshops (ICDCSW), 2013 IEEE 33rd International Conference* on (pp. 226-231).
- [27] Fister Jr, I., Fister, D., & Yang, X. S. (2013). A hybrid bat algorithm. arXiv preprint arXiv:1303.6310.
- [28] Wang, G., Guo, L., Duan, H., Liu, L., & Wang, H. (2012). A bat algorithm with mutation for UCAV path planning. *The Scientific World Journal*, 2012.
- [29] Yang, X. S. (2009). Harmony search as a metaheuristic algorithm. In *Music-inspired harmony search algorithm* (pp. 1-14). Springer Berlin Heidelberg.
- [30] Weyland, D. (2010). A rigorous analysis of the harmony search algorithm: How the research community can be misled by a "novel" methodology. *International Journal of Applied Metaheuristic Computing (IJAMC)*, 1(2), 50-60.
- [31] Wang, G., & Guo, L. (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization. *Journal of Applied Mathematics*, 2013.
- [32] Kumar, V. S., & Aramudhan, M. (2014). Trust based resource selection and list scheduling in cloud computing. *International Journal of Advances in Engineering & Technology*, 6(6).