iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 6 **Issue:** I **Month of publication:** January 2018

DOI: http://doi.org/10.22214/ijraset.2018.1101

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

Novel Approach for Work Flow Scheduling In Cloud Environment by Convex Optimization

Kiran[1], Sanjay[2]
[1]*Scholar, Department of computer science, Technical University, Hamirpur, Himachal Pradesh, India.*
[2]*Head of computer science, SIRDA institution of engineering technology, Sundernagar, (H.P), India.*

Abstract: Many algorithms in the literature have been targeting the work flow scheduling problem, however, handful efficient solutions have been proposed. This paper proposed an ACO (Ant colony optimization) algorithm based on the real ant behaviour and it's a new heuristic algorithm to solve combinational optimization problems. We use intelligence optimization ant colony optimization. This is initialized by Pareto distribution. ACO is used to converge the decision of Virtual Machine (VM) migration by its convergence to minima of cost and time. Workflow scheduling is the most important part of cloud computing, because based on the different criteria it decides cost execution and time execution. ACO performs better in comparison to GA for reduction of cost and time because of the random crossover.
Keywords: Cloud computing, ACO, intelligence optimization, Resource scheduling, VM

I. INTRODUCTION

These days, Cloud computing is a developing zone in distributed computing that conveys progressively versatile administrations on demand over the web through the equipment and programming virtualization. The greatest preferred standpoint of the cloud is its adaptability to rent and discharge resources according to the client necessity. Besides, the cloud supplier offers two sorts of plans to be specific on demand short-term plan and long-term reservation plan. It has good framework i.e. Scalability, Transparency, Security and Monitoring [1].

II. APPROACH

For the implementation, we have used the tools java, programming language JDK 1.8, eclipse. Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used java IDE.

III. IMPLEMENTATION AND RESULT

A. Basic ACO Algorithm

The Ant algorithm was introduced by Dorrigo M. in 1996 based on the real ant behaviour and it's a new heuristic algorithm to solve combinational optimization problems. Investigations have shown that ants have the ability to find food in an optimal path between the food and nest. With the ant motion some pheromone is released on ground, previous laid trail is encountered by isolated ant is being detected and follow with a higher probability. The ant's probability of choosing the way depends upon the pheromone concentration on that way. Higher the pheromone concentration, higher will be the probability of that way adoption. An optimal way can be found by utilizing this positive mechanism of feedback. The steps of ant colony optimization algorithm are given below:

1) *Algorithm 1:*
a) *Step 1:* Parameters is set, pheromone trails are initializing.
b) *Step 2:* On path segments, the virtual trail is accumulated.
c) *Step 3 :* ACO- Construct ant solutions

From node i to node j an ant will move with probability

$$P_{i,j}=\left(\frac{(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}{\sum(\tau_{i,j}^{\alpha})(\eta_{i,j}^{\beta})}\right)$$

Where,
On edge i ,j the amount of pheromone is Ti,j
To control the influence of Tijα is a parameter
In edge i,j (typically 1/di j) n i j is the desirability
To control the influence of n i j β is a parameter

d) *Step 4:* ACO –Pheromone update

According to the equation amount of pheromone is updated

T ij=(1-p)Tij$+\Delta\tau_{i,j}$

According to the equation amount of pheromone is updated

$\tau_{i,j} = (1-\rho)\tau_{i,j} + \Delta\tau_{i,j}$

Where,

On a given edge I ,j the amount of pheromone is $\boldsymbol{\tau_{i,j}}$

ρ is the rate of pheromone evaporation is ρ

The amount of pheromone deposited is $\Delta\tau_{i,j}$, typically given by

$$\Delta\tau_{i,j}^{k} = \begin{cases} \frac{1}{L_k} & \text{if ant k travels on edge i, j} \\ 0 & \text{otherwise} \end{cases}$$

Where,

The cost of the k^{th} ant's tour (typically length) is L_k.

B. Proposed Workflow Scheduling ACO Algorithm

With the above given ant algorithm characteristic utilization, the task can be scheduled. Similarly, new task can be carried out with the utilization of previous task scheduling result. The basic ideas of ACO algorithm is inherited in Workflow Scheduling-ACO algorithm for the reduction of execution time and cost.

Firstly, input the workflow to the workflow simulator and parse the task from this workflow. Pareto distribution is followed by the task. On VMs, there is a pare to distribution of ants at the beginning and then, VM_i pheromone values are initialized:

$\boldsymbol{\tau_i(0) = p_NUM_i \times p_MIPS_i + VM_b_i}$Where

p_NUM_i←Number of VM_i processor

p_MIPS_i←Million instructions per second of each VM_i processor

VM_b_i←VM_i communication bandwidth ability

Choosing VMs rule for next task: For next task, VM_i choose by k-ant with probability defined as:

$$\boldsymbol{P_i^K(T) = \begin{cases} \frac{[\tau_i(T)]^{\alpha}[c_i]^{\beta}[lb]^{\gamma}}{\sum[\tau_K(T)]^{\alpha}[c_K]^{\beta}[lb]^{\gamma}} & \text{if } i\epsilon 1,2 \ldots \ldots \ldots .\, n \\ 0 & \text{otherwise} \end{cases}}$$

Where

$\tau_i(T)$←Pheromone value of VM_i at time T

c_i←VM_i computing capacity

C_i can be defined as:

$\boldsymbol{c_i = p\_NUM_i \times p\_MIPS_i + VM\_b_i}$

lb_i←VM_i load balancing factor for minimizing the degree of imbalance defined as:

$$\boldsymbol{lb_i = 1 - \frac{et_i - Avg\_et}{et_i + Avg\_et}}$$

Where

Avg_et←Virtual machine average execution time in the optimal path last iteration

et_i←Expected execution time of VM_i task

$et_i$is defined as:

$$\boldsymbol{et_i = \frac{total\_TL}{c_i} + \frac{Input\_FS}{VM\_b_i}}$$

Where

$total\_TL$←Total length of task submitted to VM_i

$Input\_FS$←Task length before execution

α, β and γ ← Parameters controlling the relative weight of pheromone trail along with VMs computing capacity and load balancing. Once heavily loaded are some VMs becoming bottleneck in cloud influences the given task set make span. The load balancing factor lb_i is defined in the ant algorithm for improving the capacity of lead balancing. Bigger the lb_i, higher will be the probability of choosing means VM_i comprehensive ability is greater now.

Updating Pheromone: Ant Let $\tau_i(T)$ at any time T be the VM_i pheromone intensity. The update of the pheromone is given by:

$$\tau_i(T+1) = (1-\rho) \times \tau_i(T) + \Delta\tau_i$$

Where

$\rho \epsilon [0,1]$ ← Decay coefficient of pheromone trail

The past solution impact will be less if value of ρ is greater. The $\Delta\tau_i$ value is defined as:

After the completion of ant tour, updating the local pheromone on VM visited and $\Delta\tau_i$ value is given as:

$$\Delta\tau_i = 1/t_{iK}$$

Where

t_{iK} ← K-ant searched shortest path length at i^{th} iteration

In case, the current optimal solution is found by the ant while completing its tour, larger intensity pheromone is laid on its tour and updating the global pheromone on VM visited and $\Delta\tau_i$ value is given as:

$$\Delta\tau_i = d/t_{op}$$

Where

t_{op} ← Current optimal solution

d ← encouragement coefficient

If the function is optimized then we analysis the cost and time of that function.

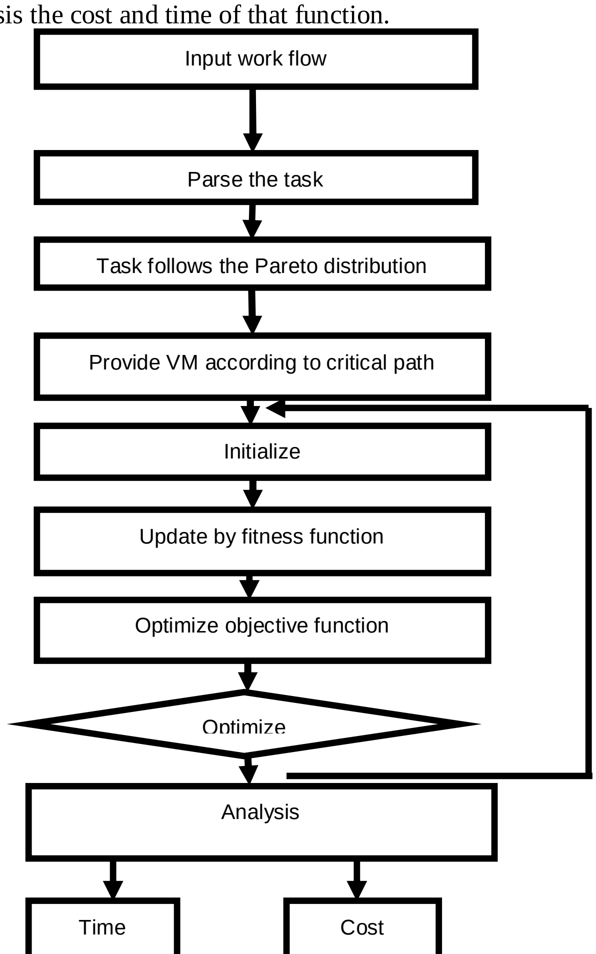


Fig.4.1.System Model

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor :6.887
Volume 6 Issue I, January 2018- Available at www.ijraset.com

IV. EXPERIMENTAL RESULT

Result of GA and ACO using –SIPHT						
Ensemble	GA			ACO		
Size	TET	TEC	Response Time	TET	TEC	Response Time
2	24.06	5723.855	0.01543485	4.59	4584.063	0.03872953
4	40.16	10011.35	0.02542084	11.26	8605.69	0.06121714
6	39.04	13521.19	0.0222077	15.13	12200.43	0.05127205
8	55.98	13404.39	0.02757845	32.83	14277.19	0.05764191
10	89.61	17549.09	0.02727286	27.77	16900.17	0.0473407
12	64.5	20561.6	0.02593575	34.34	19939.58	0.04683594
14	54.64	21713.64	0.02712497	34.06	22367.37	0.05854424
16	61.21	25855.99	0.02958881	53.59	30163.12	0.06301173
18	93.3	33493.56	0.03669985	74.36	33303.33	0.05073513
20	71.03	34154.99	0.03308249	60.55	33057.34	0.05341777

Table 4.1: Comparison table of GA and ACO using SIPHT

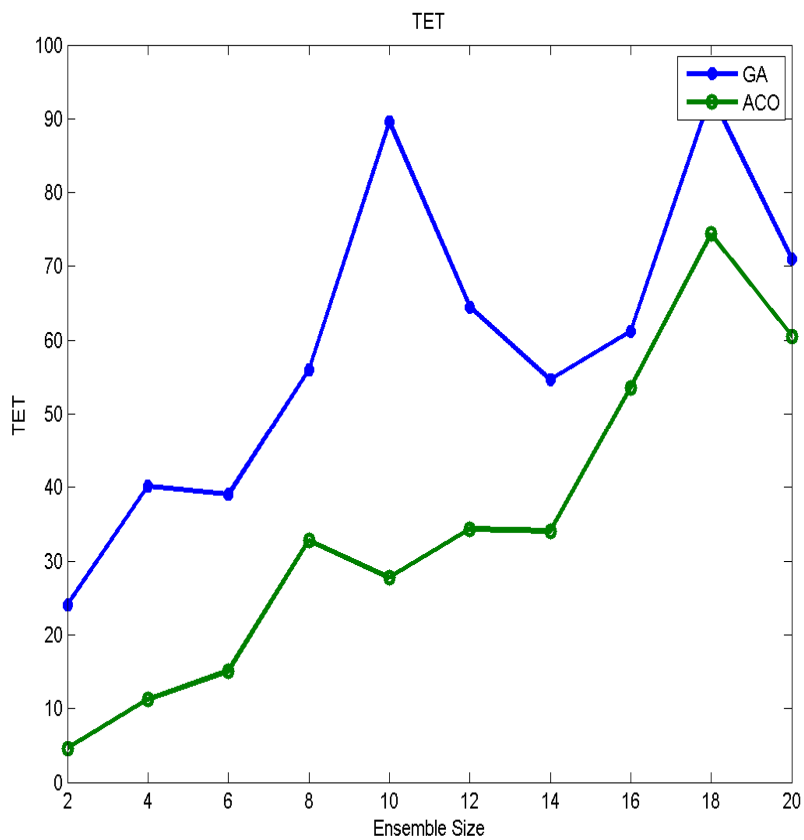


Figure4.2: Comparison graph of TET of GA and ACO using SIPHT

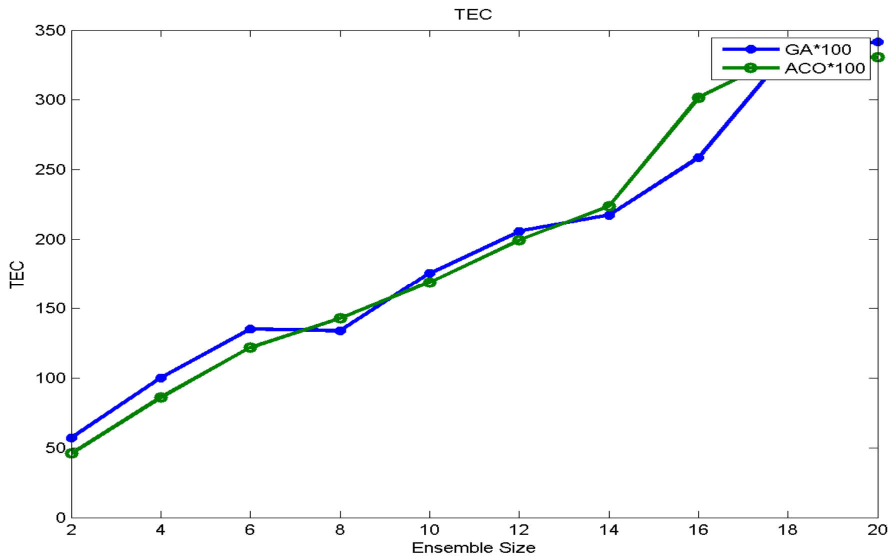


Figure4.3: Comparison graph of TEC of GA and ACO using SIPHT

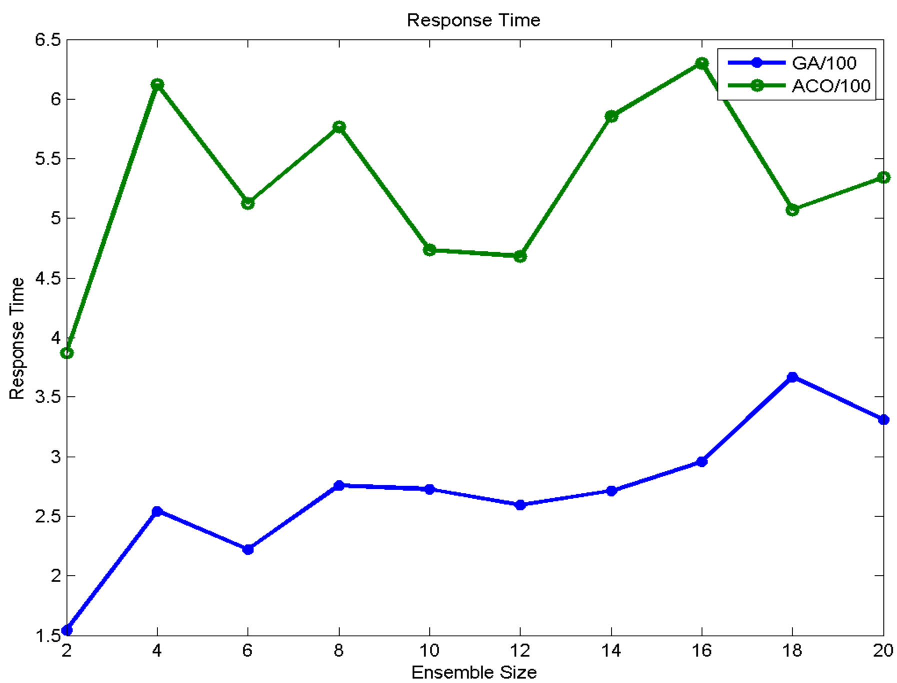


Figure 4.4: Comparison graph of Response time of GA and ACO using SIPHT

V. CONCLUSION

In this thesis works on different workflows genome, cyber shake, LIGO, SIPHT etc on TET (total execution time) and TEC (total execution cost) parameter in different virtual machine or ensemble size. In this thesis use of two to twenty ensembles size and optimize by genetic algorithm and ant colony optimization. In experiment results, In GA when given the ensemble size-4, the value of TET is 40.16, the value of TEC is 10011.35 and response time is 0.02542084. In ACO ensemble size -4, the value of TET is 11.26, TEC 8605.69 and response time 0.06121714.ACO reduces the average TET and TEC in different workflow. So, the concluded that ACO optimize and converge workflow scheduling in cloud scenario.

VI. ACKNOWLEDGMENT

I would like to express my deepest gratitude to my supervisor Mr.Sanjay for their constant guidance and encouragement, without which this work would not have been possible. For their unwavering support, I am truly grateful. I am also grateful to all my friends whose constant inspiration and support towards better work throughout my study proved to be valuable. The perfection that he brings to each and every piece of work that he does always inspired me to do things right at first time. Without his invaluable guidance, this work would never have been a successful one. He is a great person and one of the best mentors. I will always be thankful to him.

I would also like to thank the faculty of Computer Science department (SIRDA Institute of Engineering Technology).of course, this acknowledgement would not be complete without thanking my parents and family, who supported me throughout my entire study and encouraged my activities. Thanks to the loving support and faith of my parents. I was able to finish my study and make it a great time to look back on.

My gratitude is to the College for providing me assistance in the form of necessary library and laboratory facilities during the course and research work.

REFERENCES

[1] L. Liu, M. Zhang, R. Buyya. "Deadline-constrained co evolutionary genetic algorithm for scientific workflow scheduling in cloud computing." Concurrency and Computation: Practice and Experience (2016).

[2] E. N. Alkhanak, E. Nabiel, S. P. Lee, and S. U. R. Khan. "Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities." Future Generation Computer Systems 50, 3-21 (2015).

[3] S. Shukla, A. K. Gupta, S. Saxena and S. Kumar "An Evolutionary Study of Multi-Objective Workflow Scheduling in Cloud Computing." International Journal of Computer Applications (0975–8887) Volume (2016).

[4] H. Liu, D. Xu and H. K. Miao, "Ant Colony Optimization Based Service Flow Scheduling with Various QoS Requirements in Cloud Computing," 2011 First ACIS International Symposium on Software and Network Engineering, Seoul, 2011, pp. 53-58.

[5] S. Abraham, M. Naghibzadeh, and D. H. Epema, "Dead line constrained workflow scheduling algorithms for infrastructure as a service clouds," Future Generation Computer. Syst., vol. 29, no. 1, pp. 158–169, 2013..

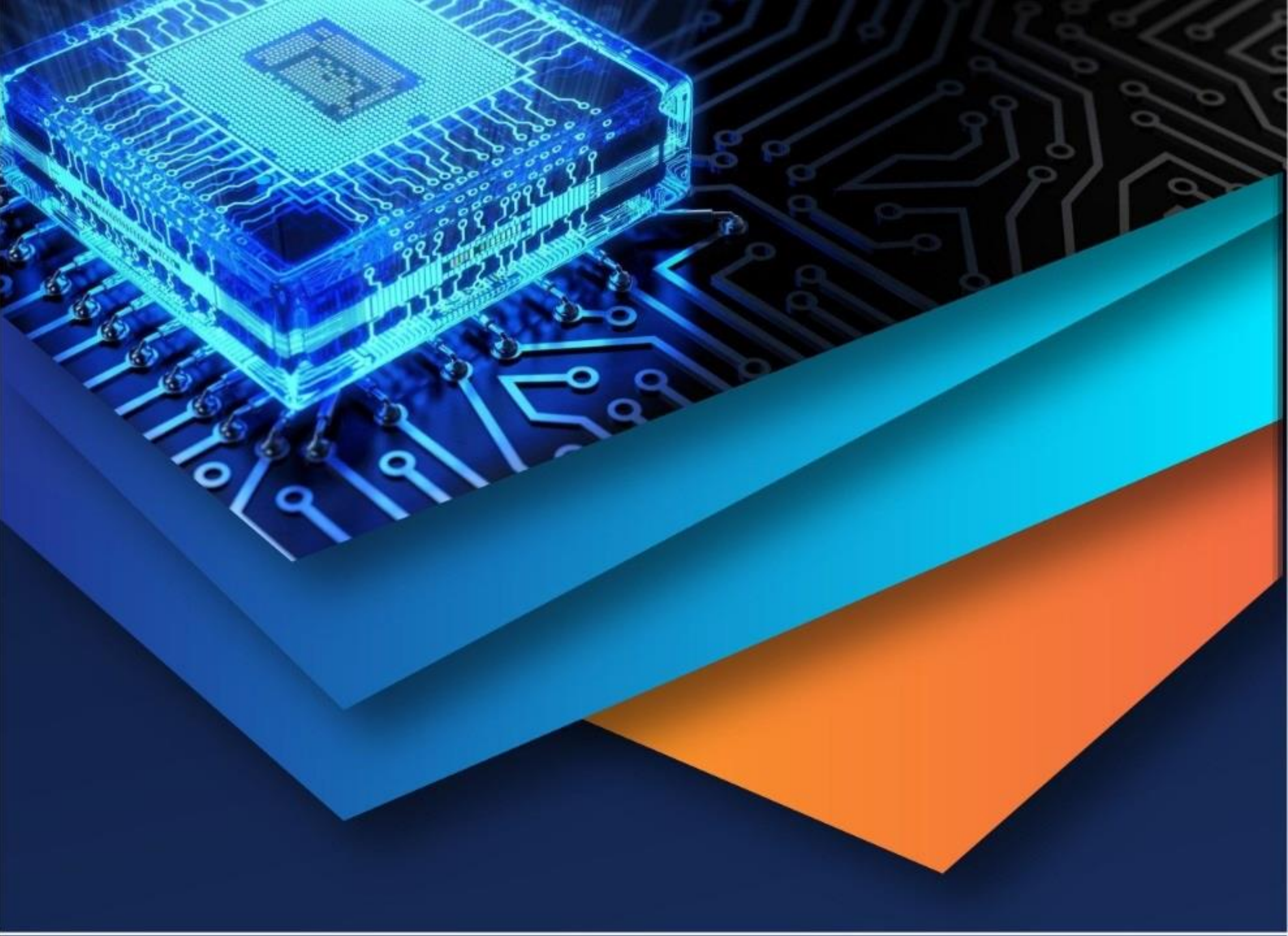
[6] A.K. Bardsiri, S.M. Hashemi, A review of workflow scheduling in the cloud computing environment, Int. J. Computer. Sci. Manage. Res. 1 (3) (2012) 348–351.

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

[8] M. Xu, L. Cui, H. Wang and Y. Bi, "A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing," 2009 IEEE International Symposium on Parallel and Distributed Processing with Applications, Chengdu, 2009, pp. 629-634.

[9] Z. Zhu, G. Zhang, M. Li and X. Liu, "Evolutionary Multi-Objective Workflow Scheduling in Cloud," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 5, pp. 1344-1357, May 1 2016.

[10] F. Zhang, J. Cao, K. Hwang, K. Li and S. U. Khan, "Adaptive Workflow Scheduling on Cloud Computing Platforms with Iterative Ordinal Optimization," in *IEEE Transactions on Cloud Computing*, vol. 3, no. 2, pp. 156-168, April-June 1 2015.



10.22214/IJRASET

INDEX COPERNICUS
45.98

IMPACT FACTOR:
7.129

IMPACT FACTOR:
7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY