Adaptive Multi-objective Task Scheduling in Cloud Computing

Ruhani Sanan[1], Pawan luthra[2], Gagandeep[3]
M. Tech Scholar[1], Assistant Professor[2], Assistant Professor[3]
Shaheed Bhagat Singh State Technical Campus of Engg. and Technology

Abstract **-** Distributed computing is a developing with worldview with extensive heterogeneous independent frameworks with adaptable computational capacity. For this planning is an essential to enhance the general calculation and increase the benefit. Cloud computing is the form of distributed computing and also a variant of grid computing. It uses highly in commercial and research purpose but one basic challenge is scheduling of the computation process. Scheduling of computation process is NP-hard problem. So effective task scheduler has adaptive sense to reduce the computation time and increase the utilization by increasing throughput. In this paper experiment performed on different optimization algorithms like BFO, ACO, and Genetic algorithm. BFO perform significant effective in throughput, energy, response time and execution time. The average improvement is 10-20% in every defined parameter.

Keywords - *BFO, ACO, Genetic algorithm, optimization*

I. INTRODUCTION

Cloud computing is an Internet-based calculation, which offers a sharing of one computer and other devices on demand, and data processing resources. Cloud computing and storage solutions for users and enterprises to provide third-party data center a variety of storage and data processing functions, these data center could away from the user - the distance around any city in a world cloud computing relies on sharing resources to ensure consistency and economies of scale, similar to utilities network electricity (such as power grid).Task scheduling important for multi-object optimization. MTS (Multi-object task scheduling) strategy is used to get the best task completion task, average cost, average energy consumption and resource utilization. In any industry, how the planning tasks of cloud resources reputed NP difficult problems of optimization, the introduction of Bacterial foraging optimization (BFO) to solve a problem of cloud computing environment task. The concept of AMTS (Adaptive multi-objective task scheduling strategy in cloud computing. The author has considered processing time and transmission time and proposed PSO (Particle Swarm Optimization) based AMTS that maximizes the resource utilization. The task scheduling is executed firstly and later, the policy of task scheduling is used for utilization of resources optimally. Better quasi-optimal solutions are obtained with respect to average energy and cost and task completion time [11]. Improved bat algorithm is proposed in this paper. This algorithm is used for load balancing and task scheduling. This algorithm also works on the pruning algorithms for population generation and then BAT is used for defining the sequence of execution of the task. The proposed method works on making the balance on the cloud and its resources and also provides the proper scheduling to all processes [2].

II. LITERATURE REVIEW

Nayak et al. [1] in this paper, the author proposed the Water Cycle Algorithm to solve the problem of scheduling in the multiprocessors. This algorithm works on the problem of heterogeneous processing environment. It is a bio-inspired algorithm which solves the problem of scheduling. In this paper different type of algorithms are compared for measuring the performance of the system. The results of the paper show that the genetic algorithm performs better.

Raj, Bibhav, et al. [2] Improved bat algorithm is proposed in this paper. This algorithm is used for load balancing and task scheduling. This algorithm also works on the pruning algorithms for population generation and then BAT is used for defining the sequence of execution of task. The proposed method works on making the balance on the cloud and its resources and also provides the proper scheduling to the all processes.

Majumder et al. [3] in this paper, the author introduces the bacterial foraging method to solve the robotic cell problem. BFO solves all the problems except the problem which belongs to the continuous domain. Pair wise interchange process of mutation is used for the tumbling and swimming operations during chemo taxis. For the random movement Cyclic Shift mutation method is used in the proposed approach. The result of the computation shows that BFO performs better than the existing methods.

Tang, Linlin, et al. [4] in this paper, the author proposed a load balancing algorithm to manage the demand of the resources and the data on the cloud. The author proposed the online and off line load balancing algorithm. BFO is used as online balancing of the data. It finishes the running task fastly and reduces the unwanted processes. The proposed method is applied on the different types of task and it performs better and provides the better efficiency.

SundarRajan et al. [5] proposed a workflow scheduling algorithm called as Firefly algorithm. This algorithm works on the behavior of fireflies. In this paper flies selects the closest possible alternative. This algorithm is used to schedule the jobs and distribute the load equally. The result of this paper show that it reduces the overall completion time in the cloud tasks.

Aron et al. [6] in this paper, the author presented an approach of secure scheduling of resources and jobs in the grid environment. Particle swarm optimization method is used for the task scheduling without violating the security rules. This algorithm reduces the execution time and increase the reliability of the systems. The results of the proposed approach show that it increases the performance and reduce the cost.

Zhang, Fan, et al. [7] Ordinal optimization is used method is used for the optimization. The working of this method is depends on the rough model. Ordinal optimization works fast and in iterative way. It maximizes the throughput of the multitasking computing. The proposed method improves the performance and reduces the cost of scheduling.

Chana et al [8] proposed the bacterial foraging method for grid resource scheduling. It is a heuristic method of scheduling. This method is used to schedule the jobs effectively in the in the grid environment. The proposed method minimizes the cost and makes span of the user applications. The result of the paper shows this technique provides the better scheduling at the low cost.

Nayak et al. [9] the author proposed a solution for the dynamic task scheduling in the multiprocessors. It solves this problem by using the hybrid optimization method. In this approach the Genetic algorithm is combined with BFO. For assigning the task to the processor GA-BF algorithm is used. The results of the proposed experiment show that it provides very effective scheduling.

Jain, Arvind Kumar, et al [10] proposed a method to reduce the transmission congestion by using the optimal bidding method. This strategy works on the bi-level optimization. This approach provides a global solution for the congestion problem. The result of the paper shows the effectiveness of the proposed methodology.

Gerkey et al. [12] studied the taxonomy of task allocation in multi-robot system. In this paper, author discussed the various architecture of the MRTA. Problems and issues related to this method are also explained in detail in this paper.

Rajni, et al. [23] in this paper, the author introduced the BFO method to solve the problem of Job scheduling. It manages the resources into the grids. The experiment result of the proposed approach shows effective and efficient result in scheduling process.

III. PROPOSED METHODOLOGY

This part of the paper explains the methodology of the work and algorithm used in it. In proposed work optimization technique is used in which Bacterial foraging optimization algorithm that solves issues related to the real-world that arising in many application domains.

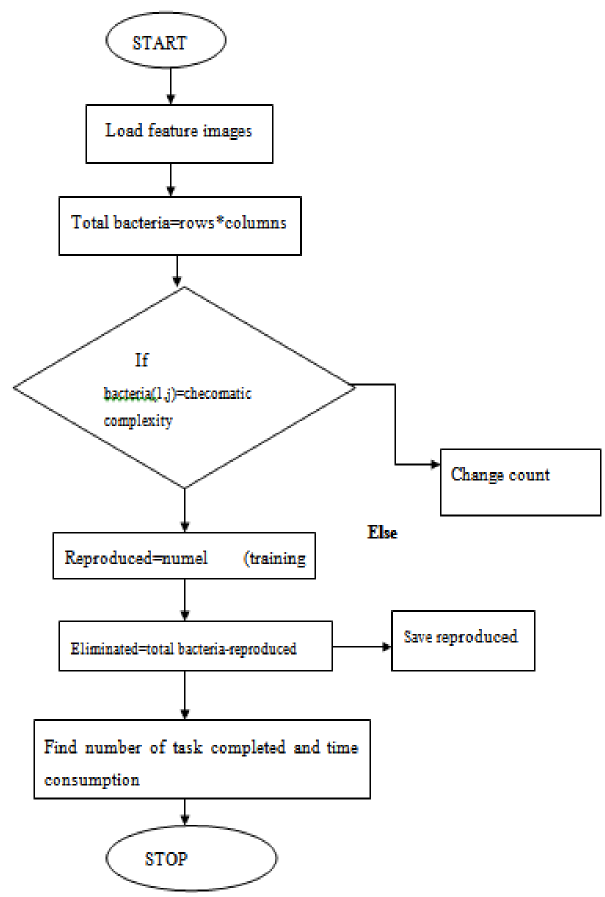

Fig.1 Flow chart of the proposed methodology

IV. METHODOLOGY

Following are the steps that are used in the methodology of the proposed work.

Step1: Start
Step2: Initially we upload feature images.
Step3: Find total bacteria
Total bacteria=rows*columns
Step4: Check bacteria complexity and training features
Step5: Find reproduction
Reproduction= numel(training features)
Step6: Start elimination
Elimination=total bacteria-reproduced
Step7: Save reproduced
Step8: In the end, evaluate results using given parameters such as: Number of task completed and time consumption.
Step9: END.

Bacterial Foraging Optimization Algorithm (Bfo)

BFO optimization algorithm is mainly used to reduce the noise, selects the features and it provides the maximized accuracy rate. It mainly works as a feature selection algorithm and it increases the processing speed, enhances the performances and improves the identification rate.

Bacterial foraging optimization algorithm inspired by bacteria such as E. coli and the primates of feeding behavior. Specifically, inspired BFOA bacterial chemotactic behavior, the bacteria will perceive the chemical environment (such as nutrients) gradient and moving to or from a particular signal [13].Bacterial products chemical gradient-based in their perceived environment branch of the food. Similarly, attraction and repulsion bacteria secret chemicals into the environment, and can be seen in a similar way to the other. The mechanisms use for the association (such as the flagella)of bacteria can move in their environment, sometimes chaotic movement (tumbling and rotation), and other times of directional way to move, it can be entered a swim. According to the cell - cell interactions, clumps of cells can be a source of food, and / or may actively reject or ignore. Motion can be achieved by stretching the fungal whole forging process of flagella. E. coli flagella helps fall or swimming, these are two basic operations in the case of foods by bacteria. When they turn in clockwise flagella, flagella fired from each cell. This outputs in the displacement of flagella respectively, and finally, with a smaller amount of roller bearing bacteria, while in the destructor, repeated gradient of hilly nutrient to be found. Bacterial flagella help noise counter clockwise to a swim very fast.

Algorithm for BFO in Multi-object task scheduling

Step1 : Initialize the parameters.
Step2 : Load feature images
Step3 : Find total bacteria=rows*columns
Step4 : Reproduction of features
Step5 : Elimination=total bacteria-reproduced
Step6 : Stop

Algorithm of Bacterial Foraging Optimization (BFO).

Step 1: Begin
Initialize the parameters, K(r), r = 1, 2, 3….A and all counter to zero.
REPEAT
For p=1 to M_{jz}
For c= 1 to M_{ij}
For e = 1 to M_k
For r = 1 to A
Evaluate E(r, e, c, p)
Then let E(r, e, c, p) = E(r, e, c, p) + E_{kk} (θ^e(e, c, p), P(e, c, p))
E_{last} = E(r, e, c, p)
Tumble: Create a random vector $\Delta(i) \in I^p$
Move : $\theta^r(e + 1, c, p) = \theta^r (e, c, p) + K(r) \frac{\Delta(i)}{\sqrt{\Delta^X(r)\Delta(r)}}$
Evaluate: E(r, e+1, c, p)
Let E(r, e+1, c, p) = E(r, e+1, c, p) + E_{kk} (θ^r(e + 1, c, p), P(e + 1, c, p))
N= 0
While n < M_a
n = n+1
IF, E(r, e+1, c, p) < E_{last}
E_{last} = : E(r, e+1, c, p)
Move : $\theta^r(e + 1, c, p) = \theta^r (e+1, c, p) + K(r) \frac{\Delta(i)}{\sqrt{\Delta^X(r)\Delta(r)}}$
Utilize this θ^r (e + 1, c, p) to evaluate new E(r, e+1, c, p) by using cell-to-cell attraction effect.
ELSE
n = M_a
ENDIF
ENDWHILE
ENDFOR
Step 2: For i = 1 to A
Evaluate $E^r_{health} = \sum_{e=1}^{M_j+1} E\ (r, e, c, p)$
ENDFOR
Sort bacteria in order of cost value of E_{health}
Terminate A_i bacteria with the maximum value of E_{health} (i.e. least healthy area)
Break every individual bacterium with the minimum value of E_{health} in two and every individual pair resides in the same original location of the parent.
ENDFOR
FOR r = 1 to A
Terminate and spread individual bacterium with probability P_{jz}, Taking Bacteria population constant.
ENDFOR
ENDFOR
UNTIL elimination criteria satisfied

END

V. RESULTS

The below given graph show the performance parameters and the variations between them. It shows BFO performs better than the other two methods.

Table 1.1 Table of Algorithms values.

Parameters	Genetic Algorithm	ACO	BFO
Throughput	38.03	40.15	75.59
Response Time	75	85.71	70.23
Execution Time	13.14	12.45	6.61
Energy Consumption	11.18	10.56	5.31

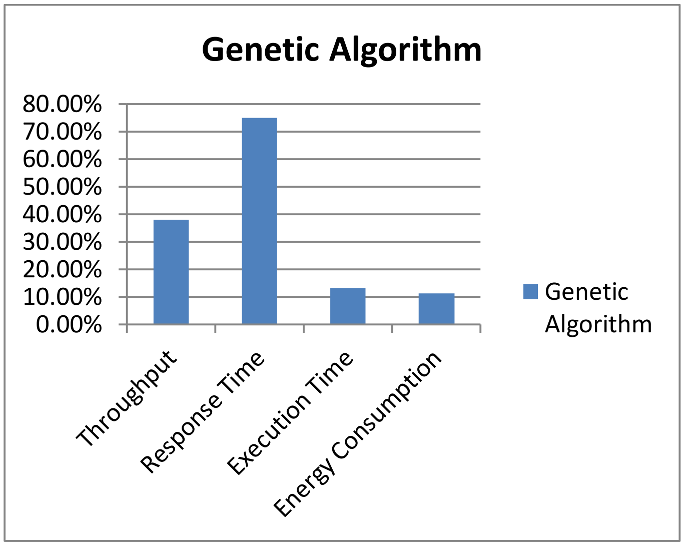

Fig.2 Performance parameters of Genetic Algorithm

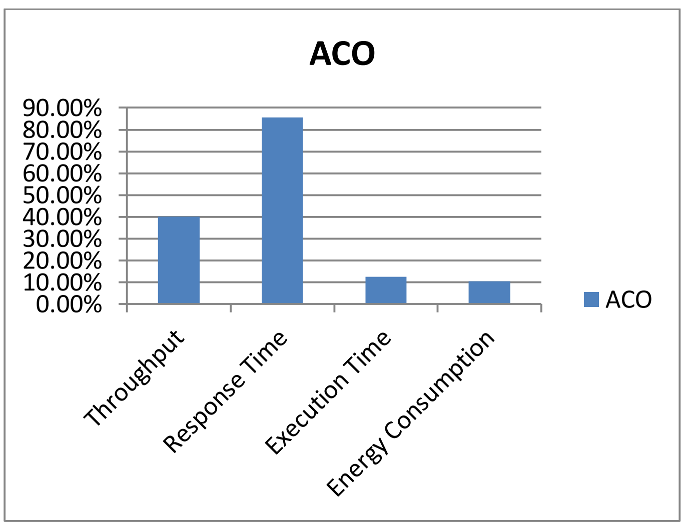

Fig.3 Performance parameters of ACO

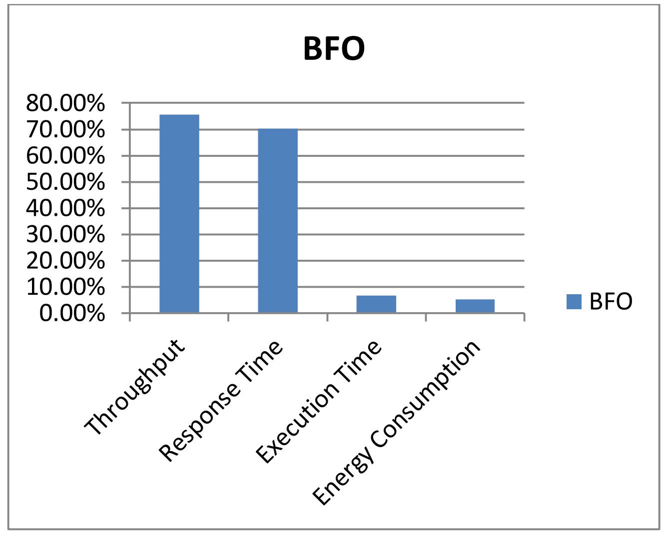

Fig.4 Performance parameters of ACO

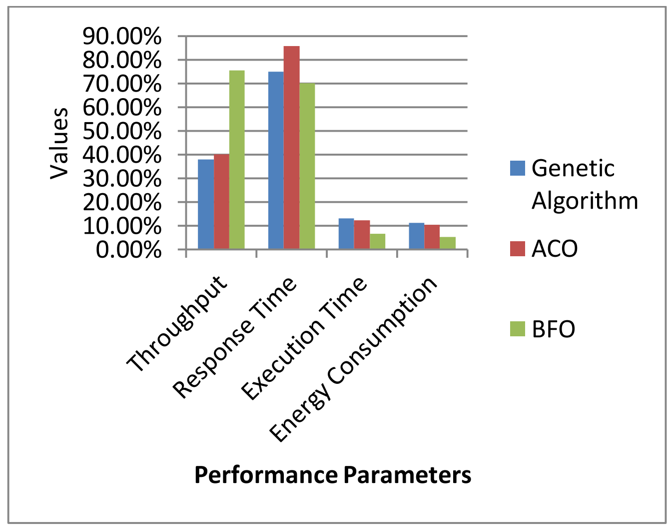

Fig.5 Comparisons of Genetic Algorithm, ACO and BFO.

VI. CONCLUSION

In this paper, we process the images in the cloud environment for the resources utilization. Because if the resources are utilized properly it is possible to handle the big task also, so this paper address the scheduling problem which adaptively schedules by the optimization algorithm. In experiment BFO show throughput 75.59 images/ sec, response time 70.23ms, execution time 6.61 sec and energy consumption 5.31KJ.So we can conclude Bacterial foraging optimization method improve parameters compare to ACO and genetic algorithm.

VII. REFERENCES

[1]. Nayak, Sasmita Kumari, Sasmita Kumari Padhy, and Chandra Sekhar Panda. "Efficient Multiprocessor Scheduling Using Water Cycle Algorithm." *Soft Computing: Theories and Applications*. Springer, Singapore, 2018. 559-568.

[2]. Raj, Bibhav, et al. "Improvised Bat Algorithm for Load Balancing-Based Task Scheduling." *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications*. Springer, Singapore, 2018. 521-530.

[3]. Majumder, Arindam, and Dipak Laha. "Bacteria Foraging Optimization Algorithm for RoboticCellScheduling Problem." *Materials Today: Proceedings* 4.2 (2017): 2129-2136.

[4]. Tang, Linlin, et al. "Online and offline based load balance algorithm in cloud computing." *Knowledge-Based Systems*138 (2017): 91-104.

[5]. SundarRajan, R., V. Vasudevan, and S. Mithya. "Workflow scheduling in cloud computing environment using firefly algorithm." *Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on*. IEEE, 2016.

[6]. Aron, Rajni, Inderveer Chana, and Ajith Abraham. "A hyper-heuristic approach for resource provisioning-based scheduling in grid environment." *The Journal of Supercomputing* 71.4 (2015): 1427-1450.

[7]. Zhang, Fan, et al. "Evolutionary scheduling of dynamic multitasking workloads for big-data analytics in elastic cloud." *IEEE Transactions on Emerging Topics in Computing* 2.3 (2014): 338-351.

[8]. Chana, Inderveer. "Bacterial foraging based hyper-heuristic for resource scheduling in grid computing." *Future Generation Computer Systems* 29.3 (2013): 751-762.

[9]. Nayak, Sasmita Kumari, Sasmita Kumari Padhy, and Siba Prasada Panigrahi. "A novel algorithm for dynamic task scheduling." *Future Generation Computer Systems* 28.5 (2012): 709-717.

[10]. Jain, Arvind Kumar, et al, “Bacteria foraging optimization based bidding strategy under transmission congestion,” IEEE Systems Journal 9.1 (2015): 141-151.

[11]. Kim, Joo-Young, et al, “A 201.4 GOPS 496 mW real-time multi-object recognition processor with bio-inspired neural perception engine,” IEEE Journal of Solid-State Circuits 45.1 (2010): 32-45.

[12]. Gerkey, Brian P., and Maja J. Mataric, “A formal analysis and taxonomy of task allocation in multi-robot systems,” The International Journal of Robotics Research 23.9 (2004): 939-954.

[13]. Bowman-Amuah, Michel K, “Multi-object identifier system and method for information service pattern environment,” U.S. Patent No. 6,539,396. 25 Mar. 2003.

[14]. Wu, Chunguo, et al. "Improved bacterial foraging algorithms and their applications to job shop scheduling problems." *International Conference on Adaptive and Natural Computing Algorithms*. Springer, Berlin, Heidelberg, 2007.

[15]. Schäfer, Jan, and ArndPoetzsch-Heffter, “JCoBox: Generalizing active objects to concurrent components,” European Conference on Object-Oriented Programming. Springer Berlin Heidelberg, 2010.

[16]. Kołodziej, Joanna, and Fatos Xhafa. "Meeting security and user behavior requirements in Grid scheduling." *Simulation Modelling Practice and Theory* 19.1 (2011): 213-226.

[17]. Aron, Rajni, Inderveer Chana, and Ajith Abraham. "Hyper-heuristic based resource scheduling in grid environment." *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on*. IEEE, 2013.

[18]. Dutta, Maitreyee, and Naveen Aggarwal. "Meta-Heuristics Based Approach for Workflow Scheduling in Cloud Computing: A Survey." *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, New Delhi, 2016. 1331-1345.

[19]. Li, Jun, Chunlin Li, and Qingqing Li. "A research about independent tasks scheduling on tree-based grid computing platforms." *Intelligent Systems and Applications (ISA), 2010 2nd International Workshop on*. IEEE, 2010.

[20]. Sathya, P. D., and R. Kayalvizhi. "Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images." *Measurement* 44.10 (2011): 1828-1848.

[21]. Mani, V., Sundaram Suresh, and H. J. Kim. "Real-coded genetic algorithms for optimal static load balancing in distributed computing system with communication delays." *International Conference on Computational Science and Its Applications*. Springer, Berlin, Heidelberg, 2005.

[22]. Fan, Zongqin, et al. "Simulated-annealing load balancing for resource allocation in cloud environments." *Parallel and Distributed Computing, Applications and Technologies (PDCAT), 2013 International Conference on*. IEEE, 2013.

[23]. Rajni, Inderveer Chana. "Resource provisioning and scheduling in Grids: issues, challenges and future directions." *Computer and Communication Technology (ICCCT), 2010 International Conference on*. IEEE, 2010.

[24]. Agrawal, Vivek, Harish Sharma, and Jagdish Bansal. "Bacterial foraging optimization: A survey." *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011) December 20-22, 2011*. Springer Berlin/Heidelberg, 2012.

[25]. Balázs, Krisztián, Zoltán Horváth, and László T. Kóczy. "Hybrid bacterial iterated greedy heuristics for the permutation flow shop problem." *Evolutionary Computation (CEC), 2012 IEEE Congress on*. IEEE, 2012.