IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-2, Feb 2019 E-ISSN: 2347-2693

# Optimized Scheduling Procedure for Enhancing Resource Utilization in Hetrogeneous Cloud Enviornment

**Hardeep Kaur[1*], Anil Kumar[2]**

[1,2]Department of Computer Science and Technology, Guru Nanak Dev University, Amritsar, India

[*]*Corresponding Author: badhanhardeep7@gmail.com 9876871559, 9501033889*

DOI: https://doi.org/10.26438/ijcse/v7i2.207215 | Available online at: www.ijcseonline.org

Accepted: 16/Feb/2019, Published: 28/Feb/2019

***Abstract:*** Because of a phenomenal increment in the quantity of computing assets in various associations, compelling jobs scheduling algorithms are required for proficient asset use. Job scheduling in considered as NP difficult issue in parallel and disseminated registering situations, for example, group, matrix and mists. Meta-heuristics, for example, Genetic Algorithms, Ant Colony Optimization, Artificial Bee Colony, Cuckoo Search, Firefly Algorithm, Bat Algorithm and so on are utilized by researchers to get close ideal answers for work scheduling issues. These meta-heuristic algorithms are utilized to plan distinctive sorts of jobs, for example, BSP, Workflow and DAG, Independent undertakings and Bag-of-Tasks. This paper is an endeavor to give exhaustive review of prominent nature-enlivened meta-heuristic procedures which are utilized to plan distinctive classifications of jobs to accomplish certain execution targets.

***Keyword***: ACO, BAT, Cuckoo, genetic algorithm

## I. INTRODUCTION

Cloud computing in modern era provides way of using resources without their physical presence at source[1]. The service provided by the cloud is at the front end of computing and internet is at back end. In other words internet is heart and soul of cloud computing. Cloud computing provide mechanism for the users to perform operations that required heavy resources not possessed by them at pay per use basis. With the rapid development of hardware and software cloud computing brings the revolution in the business industry. It provides resources like computational power, storage, computation platform ad applications to user on demand through internet. Some of the cloud providers are Amazon, IBM, Google, Sales force, Microsoft etc. Cloud computing features included resource sharing, multi-tenancy, remote data storage etc. but it challenges the security system to secure, protect and process the data which is the property of the individual, enterprises and governments[2]. Even though, there is no requirement of knowledge or expertise to control the infrastructure of clouds; it is abstract to the user. It is a service of an Internet with high scalability, quality of service, higher throughput and high computing power. Cloud computing providers deploy common online business applications which are accessed from servers through web browser. Data security is the biggest issue in cloud computing and it is not easy to resolve it[3]. In our review paper we will review the different ways to manage the confidentiality of the data. Before discussing migration mechanisms we discuss services provided by cloud along with types of cloud.

### 1.1 Cloud Services

There exist legion of services associated with cloud. These services are as described below

- IaaS

Infrastructure as services is critical services provided through cloud virtualized computing resources are provided by the application of IaaS. Internet is key element with which IaaS is accessed. Cast is encountered on the basis of usage. [4]

- PaaS

Platform as a service is another cloud service that enhances the organizational applications. Large number of applications exits that are supposed to execute over the distinct machines. All the applications has distinct requirements in terms of platform. This platform requirement is accomplished using cloud computing. Cost is encountered on the basis of time period for which platform is online[5].

- SaaS

Software as a service is another critical service supported through cloud. Cloud computing host software which can be accessed by users having access to cloud. In other words machines having limited resources can use SaaS to access software's that they don't possess.

### 1.2 Job Scheduling

[6]Job scheduling is a procedure in which jobs apportioned

to the PC are scheduled by utilizing single cluster or multicluster approach. The job allocation process is done in light of constrained resources. In both the approaches we have different algorithms to solve the jobs like FCFS, SJF, Round Robin, and Priority Scheduling in single cluster job. Furthermore, in case of multi cluster job we can use algorithms like ACO, Firefly, Honey Bee, BAT, Cuckoo etc are used to allocate resources so that jobs can be execute in efficient manner and optimal results can be achieves in short span of time.

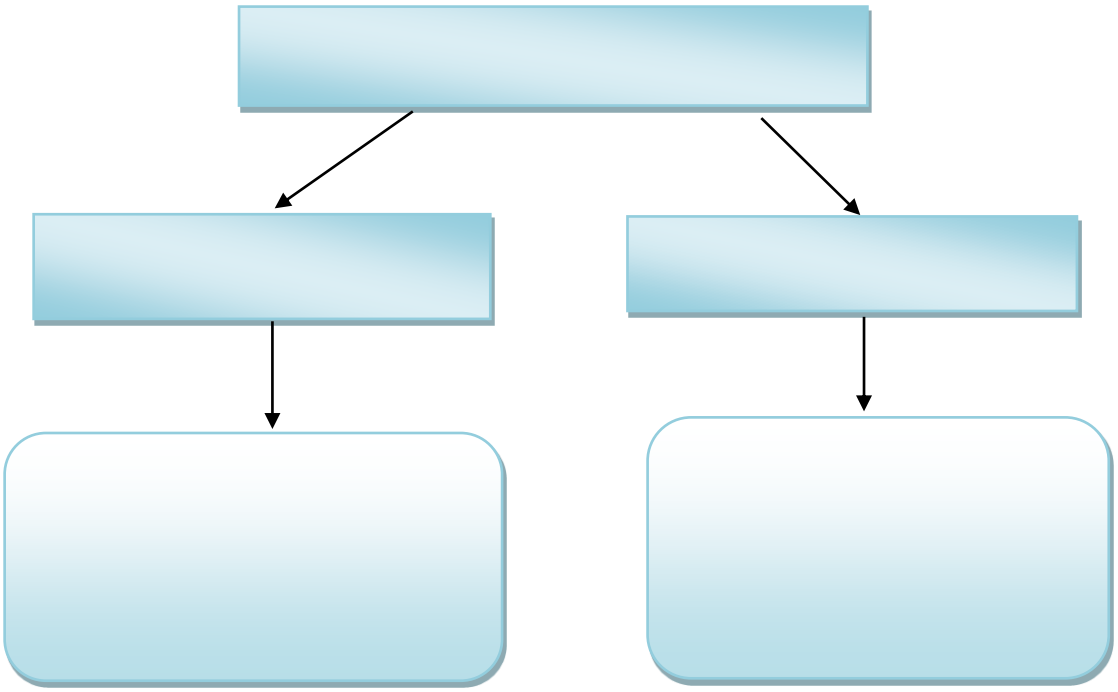


Figure 1: Scheduling in single and Multi Cluster Environment

**1.3 SCHEDULING UNDER SIMPLE ENVIORNMENT**

The multi cluster environment originates from single cluster homogeneous environment. The algorithm used under this environment includes

- FIFO Job Scheduling

[8]In this job scheduling algorithm job is distributed to the processor within the system on the first come first serve basis. This algorithm may or may not yield optimal solutions. The allocated processor release the job when the job burst time finishes. This algorithm is strictly non primitive in nature. It is hence rarely utilized in the parallel environment.

- TIME SHARING

In this scheduling time is shared among multiple jobs. The time sharing system utilizes time quantum. The processor is switched among the processors based on time quantum. The process continues until all the jobs finish execution. The time sharing system involves states such as waiting, active and ready. The time scheduling on parallel system can be implemented using local scheduling. The processing node has processors associated with them. Threads ready to be executed are placed within first come first serve buffer. [9]When the processor is available thread is fetched from the queue and executed. The rime sharing environment generally adopt pipeline concept for executing instructions concurrently within uniprocessor systems.

- GANG SCHEDULING

This scheduling produces optimal result as compared to all other parallel scheduling algorithms. This algorithm is primitive in nature. Hence deadlock never occurs within the system. The tasks that form a job are grouped together and then scheduled in this approach. The job priority is also considered in this case. The job with highest priority is executed at first place in this case. in the early system this approach produces optimal results. But nowadays more advanced algorithms exists which produces better result as compared to this algorithm.

**1.4 MULTI CLUSTER ENVIRONMENT**

Clusters are gathering of servers with distinct setup. With single cluster, hubs are of comparative arrangements. [10]As multiple clusters are bound together to form multi cluster condition. In multi cluster condition in this manner, particular nodes (servers) are assembled together. Multi cluster condition is utilized as a part of request to execute complex jobs. These jobs may not be executed by single cluster consequently distribution methodologies are constructed. Multi cluster condition is appeared as under

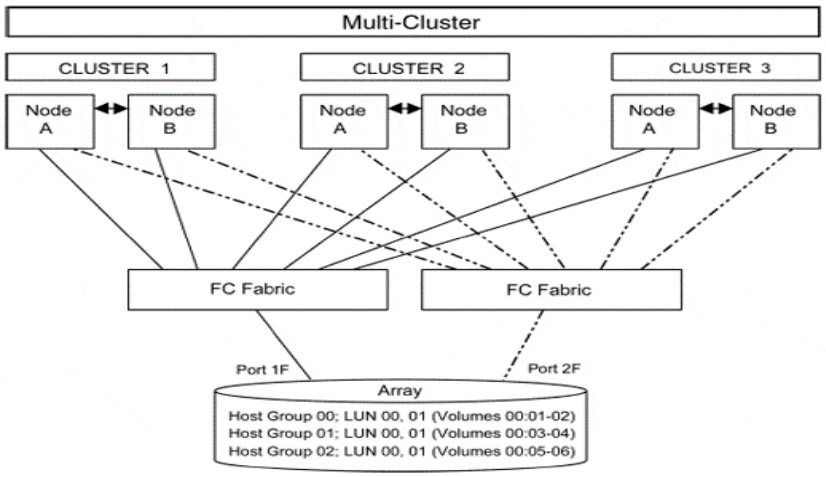



Figure 1.4.1: Multi Cluster Environment

**1.5 Parallel Computing**

Parallel computing is working on the rule that enormous issues can be subdivided into littler ones, which are then unraveled in the meantime. In this sort of calculation many jobs are executed at the same time on various processors [11]

The jobs in parallel computing can be characterized by the level at which the equipment support parallelism, with multi-center and multi-processor PCs having numerous handling components inside a solitary machine, while groups, MPPs and lattices utilize various PCs to chip away at a similar job. [12]

Particular parallel PC designs are once in a while utilized close by traditional processors, for quickening particular jobs. Parallel programming models and devices are required for elite computing. The accessibility of multi-center CPUs has given new slant to the common memory parallel programming approach. Parallel computing is operating on the rule that huge problems can frequently be subdivided into smaller ones, which are then solved at the same time. In this type of computation many jobs are executed simultaneously on different processors

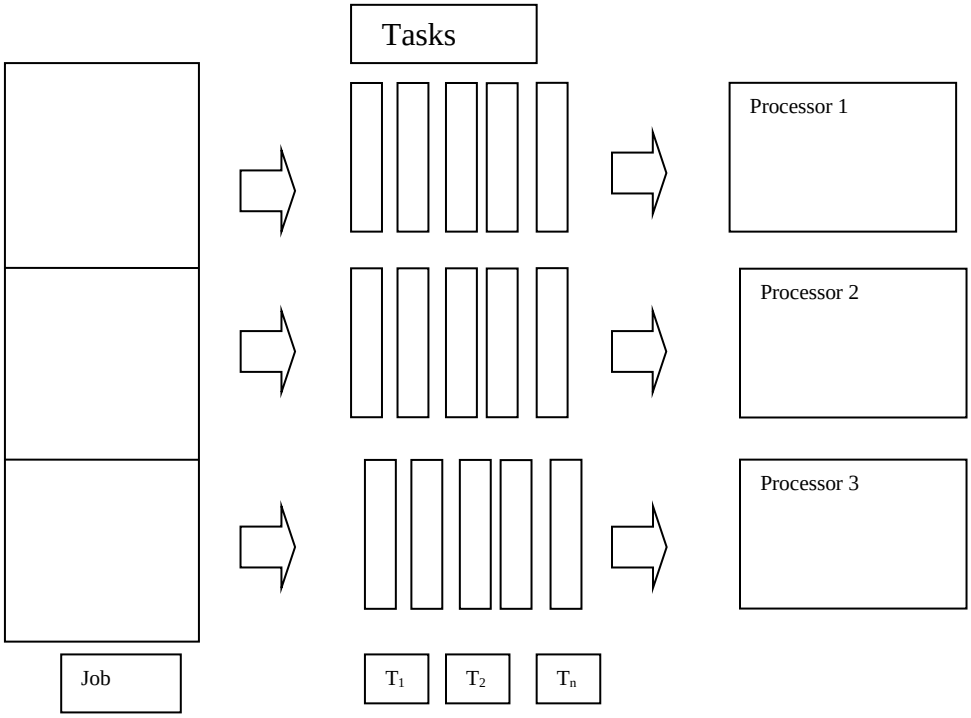



**Figure 1.5.1** Parallel Computing Of Jobs

In parallel the jobs are subdivided into tasks and after that those errands are doled out to the different resources as indicated by job necessity. The parallel job calculation is far much superior to the successive calculation in light of the fact that in consecutive calculations the jobs are executed in a solitary line to one processor. So it expends a great deal of time to execute number of jobs .So to determine this issue the jobs are executed in parallel so by doing as such the execution time of jobs are being decreased and expanded the throughput of the framework.

## II. LITERTURE SURVEY OF MULTI HEURISTIC APPROACH FOR JOB SCHEDULING

### 2.1 Multiheuristic Approach

[13]Multiheuristic approach is used to NP- hard Problems. These problems are very complex in nature which may not be sold using the single processor system. There requirements are such that single processor is unable to satisfy the requirements or need of the system. So systems with more than one processors or multi clusters are required. With every job objective is associated. These objectives are required to be satisfied in order to complete the job. Such jobs fall under the category of multiheuristic. The algorithms are as under.

### 2.1.1 GENETIC ALGORITHM

Genetic algorithm is utilized as a part of request to take care of job schedule issue in multi bunch condition. With each activity target work is related. Genetic approach continues to fulfill the target work related with the activity. This algorithm ends on the off chance that target work related with each activity is fulfilled or number of ages ends. There are stages related with the genetic algorithm.[14]

Initialization

Jobs and resources at first place required to be initialized. Jobs are known as chromosomes. These chromosomes are initially selected randomly for allocation. Generations, objective function, crossover and mutation probabilities are initialized in this phase.

Selection[15]In genetic algorithm, Jobs are referred to as Chromosomes. Resources are required by chromosomes. Chromosomes are selected initially for resource allocation. This process is known as selection. In genetic algorithm, this selection is performed using variety of mechanism known as selection criteria's. Chromosomes are selected for later breeding using this phase. Steps in selection are listed as under

- Fitness function is evaluated associated with each job. Fitness values are normalized by adding the fitness values and then dividing it with total number of jobs.
- Population is sorted according to descending value of fitness values
- Population with highest fitness value is selected for mating.

Crossover or Mating

Selected Chromosomes are then go through this phase in order to determine chromosomes which are to be mutated. Uniform crossover is preferred in proposed thesis.

Mutation

Chromosomes selected for mating in crossover are mutated to generate new chromosomes and then evaluated again.

After performing all the phases, fitness values are analyzed again. This process continues until desired level of fitness values is achieved or generations terminates.

### 2.2 ANT COLONY OPTIMIZATION

[16]This is an algorithm which is inspired from the natural behavior of ants. Ant colony optimization considers the random walk associated with Ants. Ants look for food source and if found laid down pheromone trail to be followed by other Ants. This process is simulated in job scheduling. Ants act as Jobs and Food act as resources. With Ant colony Optimization Convergence is not always guaranteed. In other words optimal solution may or may not be generated.[17] Result in terms of local and global best solution is obtained. Iterations in this algorithm continue until all the Ants gets the food source. Following steps are critical in Ant Colony Optimization

- All the Ants must visit all the resources at least once
- Distant resource is less likely to be visited due to visibility problem.
- In case resource is fond by Ant, pheromone is laid down.
- After previous iteration, Pheromone evaporated.
- Process continues until optimal solution gbest and lbest is found out or generations terminates.

Convergence problem exists while using Ant colony Optimization.

### 2.3 BAT ALGORITHM

[18]Bat Algorithm was introduced by Yang in year 2010 based on the echolocation behavior of Bats. To estimate the distance of prey, bat used echolocation to sense the distance.BAT algorithm is totally based on the behavior of Bats. The bats usually fly randomly with a frequency, loudness, pulse emission, velocity and position to search for their prey. When the bats hunt for the prey, they automatically adjust their parameters like velocity, frequency, loudness and pulse emission based on the distance among them and the prey.

[19]BAT algorithm is used in many fields like in workflow, for independent jobs and for the parallel jobs and has many applications.BAT algorithms is used for scheduling jobs among the number of processors in a given problem to minimize objective function.

By changing the velocities and the position of BATS gives the similarity of another metaheuristic known as the Particle Swarm Optimization algorithm.

[20]BAT algorithm is a metaheuristic algorithm used to allocate resources which are detected. It was inspired by echolocation behavior of BAT. Searching criteria followed is random in nature. It is particularly useful for constrained intensive environment. The balance between exploration and exploitation has to be maintained which can be accomplished through hybrid approach. Pseudo Code for BAT is as under

- Obtain Job Sequence(J1,J2,J3,-------,Jn)
- Associate Objective Function with Each Job $F(y) = \{y_1, y_2, - - - - -, y_n\}$
- Define range of BAT in terms frequency(F) and Loudness(A)
- Repeat following Steps until i<Max_Iterations
  Generate Solutions by adjusting Existing frequency
  Update location of BATS in case solution found
  If (Optimal Solution==True)
  Select this solution and replace it with existing solution
  End of if
  Randomize the BATS flying and relocate the resources.
  In case optimal solution is located increase pulse rate and decrease loudness (A).
  Rank the BATS and sort the solution.
  End of loop
- Output Solution and job Ordering(J)

**2.4 CUCKOO ALGORITHM**

[21]Cuckoo search is a meta-heuristic algorithm inspired by the bird cuckoo, these are the Brood parasites birds. It never builds its own nest and lays their eggs in the nest of other host bird nest. Cuckoo is a best-known brood parasite. Some host birds can employ directly with the intruding cuckoo. If the host bird identifies the eggs that are not their own egg then it will either throw that eggs away from its nest or simply rid its nest and build a new nest.

In a nest, each egg represents a solution and cuckoo egg represents a new and good solution. The obtained solution is a new solution based on the existing one and the modification of some characteristics. In the simplest form each nest has one egg of cuckoo in which each nest have multiple eggs represents a set of solutions. [22]CS is successfully used to solve scheduling problems and used to solve optimization problems in structural engineering. In many applications like speech reorganization, job scheduling, global optimization cuckoo search algorithm is used. Cuckoo search idealized such breeding behaviour and can be applied to various optimization problems

This algorithm is also a part of optimization algorithm which is used in order to produce optimal solution by considering decrease in time and energy consumption in multi objective scenario. The detailed steps are listed as follows:

- Initialize Cuckoo with initial set of eggs.
- Lay eggs in distinct nests.
- Some eggs are defected and killed
- If population generated is less than threshold value then check for survival of eggs in nest and obtain profit values.
- Otherwise kill cuckoo in worst area.
- If stop condition satisfied then obtain maximize solution otherwise go to step 2.

Pseudo code for Cuckoo Algorithm is described as under

- Initialize Jobs(J) for Execution along with number of resources available(R), y=0
- Input number of generations G.
- Define Objective function F(x).
- Define initial population in terms of nest
- Solution is in terms of eggs which must be assigned with rank examined through fitness values
- Repeat while stopping criteria is matched or y<G
  Y=y+1
  Produce new solution using levy flight of cuckoo
  Calculate Fitness value
  Select the nest randomly
  If ($F_i$<$F_{i+1}$)
  Replace current solution with the new solution z
  End of if
  Asses fitness and sort the solution with maximum fitness value.
  End of loop
- Produce result in terms of Job Ordering

**a. FIREFLY ALGORITHM-**

[23]Firefly Algorithm (FA) is a metaheuristic algorithm for worldwide advancement, which is roused by blazing conduct of firefly creepy crawlies. This algorithm is proposed by Xin-She Yang in 2008. Fireflies utilize the blazing conduct to pull in different fireflies, generally to send signs to inverse sex. Be that as it may, in the numerical model, utilized inside Firefly Algorithm, just the fireflies are unisex, and any firefly can pull in different fireflies.

Attractiveness of a firefly is relative to its splendor and for any couple of fireflies, the brighter one will pull in the other; so the less brilliant one is moved towards the brighter one. This is performed for any parallel mix of fireflies in the populace, on each cycle of algorithm.

[24]There has been a developing exertion in diminishing vitality utilization of substantial scale cloud server farms by means of boost of host-level usage and load adjusting methods. In any case, with the current presentation of Container as a Service (CaaS) by cloud suppliers, expanding

the use at virtual machine (VM) level winds up plainly fundamental. To this end, this paper concentrates on finding productive virtual machine sizes for facilitating holders in such a route, to the point that the workload is executed with least wastage of assets on VM level. Reasonable VM sizes for compartments are ascertained, and application assignments are gathered and grouped in view of their use designs got from authentic information. Besides, assignments are mapped to compartments and holders are facilitated on their related VM sorts. [25]We investigated mists' follow logs from Google group and consider the cloud work- stack differences, which is pivotal for testing and approving our proposed arrangements. Exploratory outcomes appeared to 7.55% change in the normal vitality utilization contrasted with benchmark situations where the virtual machine sizes are settled. In expansion, contrasting with the benchmark situations, the aggregate number of VMs instantiated for facilitating the compartments is likewise made strides by 68% overall.

Flow Time- Flow time is the average amount of time to taken by job to achieve the optimal result. Every single unit in job can be executed within that time.

$$Flow_{time} = \sum \frac{Job_{Completion_{Time}}}{Total_{jobs}} \quad (1)$$

Waiting Time – The amount of time for which job have to wait for the processor to accomplish its task.

$$waiting_{time} = Starting\_time - Arrival\_time \quad (2)$$

Execution Time- Execution time is the total amount of time to execute single job.

$$execution_{time} = finish_{\_time} - start\_time \quad (3)$$

Normalization Function- Normalization function is used in order to optimize the result which is obtained through metaheuristic algorithm.

Fitness Function- This function is used in order to analyze the fitness of the population or chromosomes. The population will alter according to this fitness function.

**2.6 Terminology Used**

Comparison table showing the analysis of algorithms used within the multi heuristic approach for future endeavors.

Table 1: Comparison of techniques used within cloud computing for scheduling

| ALGORITHM | BASIS | OPERATORS | ADVANTAGES | DISADVANTAGES |
|---|---|---|---|---|
| Genetic Algorithm (GA) [7] | Based on population size | • Crossover<br>• Mutation<br>• Selection<br>• Inversion | • Used for rule extraction<br>• for optimization problems | • Time taken for convergence is more.<br>• No guarantee for global optima. |
| BAT Algorithm[18] | Based on the echolocation behaviour of virtual bats. | • Population size<br>• Loudness parameter<br>• Pulse rate<br>• Maximum number of iterations | • Frequency tuning<br>• Automatic zooming into the region of global solution<br>• Parameter control ensures efficient exploration and exploitation. | • Convergence in case of complex problems is slow<br>• Searching in case of complex environment can be misleading. |
| ACO[16] | Based on natural behaviour of ants | • Jobs<br>• food<br>• walk | • faster convergence as compare to GA | • finding local and global conclusion can be complex |
| Shadow replication[27] | Based on number of cores associated with VMs | • Resources<br>• Cores | • Converge faster<br>• Load balancing | • Problem arises as cores fails |
| Credit based scheduling[28] | User priority and task length is considered for allocation of resources | • Resources<br>• Task length | • Make span is reduced<br>• Flow time is reduced | • Advance priority reservation is complex |

| | | | | |
|---|---|---|---|---|
| Data centre energy aware algorithm [29] | Energy of VM is considered for scheduling | • Energy conservation<br>• Execution time | • Energy is saved<br>• Execution time is reduced | • Load on VM could cause deterioration |
| Pre-emptive Scheduling[30] | Starvation problem is tackled in the allocation of resources | • Make span<br>• Flow time | • Make span and Flow time is optimised | • Priority of job is not considered for pre-emption hence critical jobs are prompted decreasing reliability |

## III. RESEARCH GAP

In existing work the genetic algorithm was utilized as a bit string representation and genetic operators. This approach is simple but it has some disadvantage that it does not applied on larger problem. The ideas of both groups and other active schedule-based GAs are suitable for middle-size problems; however, it seems necessary to combine each with other heuristics such as the shifting bottleneck or local search to solve larger-size problems. Hence to solve larger-size problems the maximum VM utilization and Firefly Algorithm (FA) are combined together.

## IV. PROBLEM DEFINITION

Grid scheduling is a NP-hard problem so soft computing techniques find the immediate application for scheduling the job on heterogeneous diversified resource environment. The constrains and parameters are selected for evaluating the fitness of each job and applying genetic operators to refine the results for optimal schedule planning. There are two disadvantages. One, every generation of the GA goes significantly more gradually (yet you require numerous less generation for the most part). Two, It lose some diversity, which can make coverage to a suboptimal arrangement all the more regularly.

Main problem with Genetic algorithm is rate of convergence that causes Make span and Flow time to increase significantly. This can be reduced by the use of hybridization of Firefly Algorithm (FA) with maximum VM utilization mechanism.

## V. OBJECTIVES

The objectives by elaborating the existing literature could be listed as under

- To reduce the cost by examining each and every available path for job execution

The path evaluation can yield large amount of paths. In case of simple firefly algorithm, only one path remain fixed and fireflies glow for the same path again and again causing static solution. In order to tackle the issue cost associated with the path is examined and optimal path can be selected corresponding to every iteration.

- To reduce the Make span associated with schedule

Make span is the time consumed corresponding to every schedule. This has to be minimized.

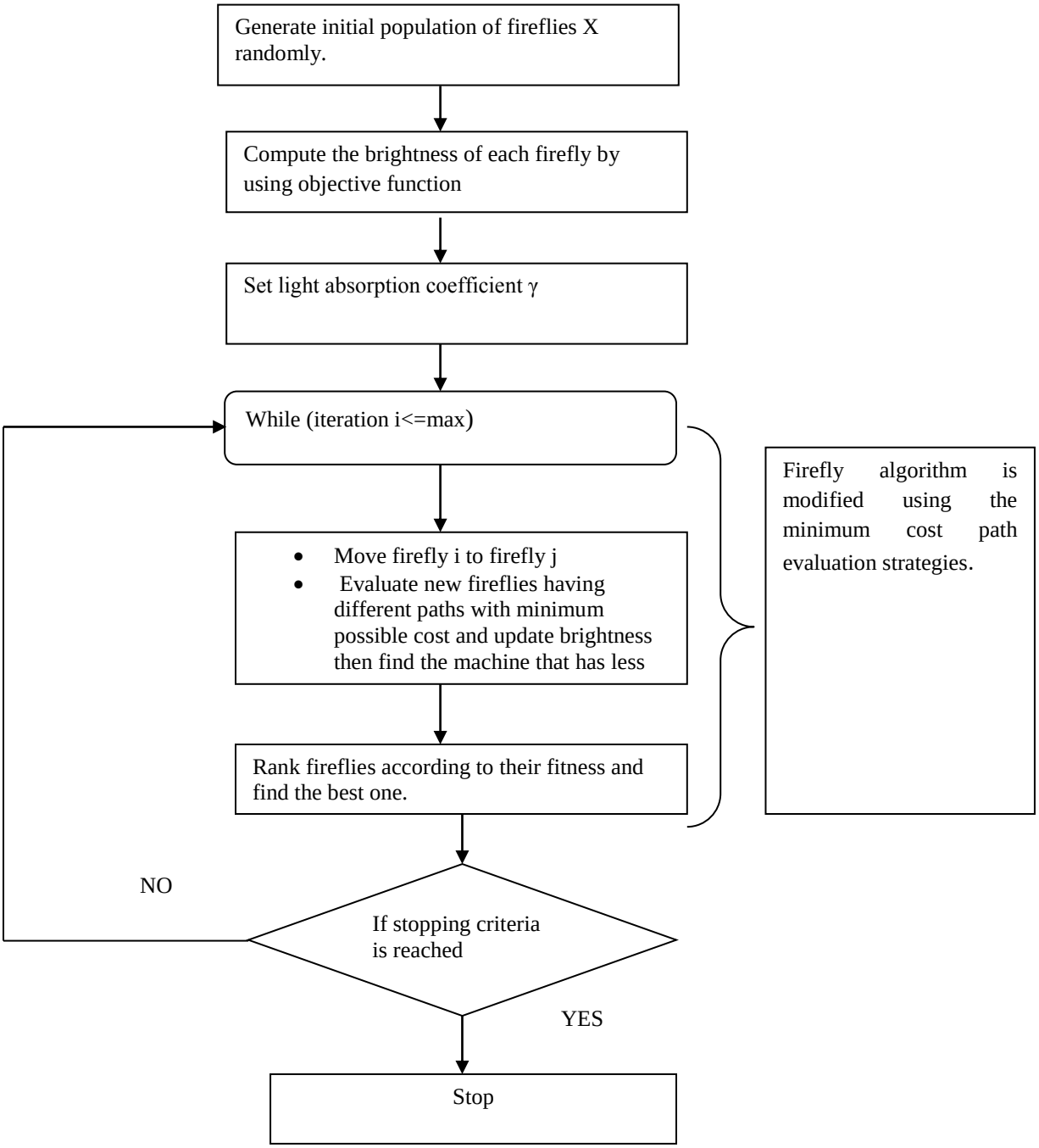
- To compare the existing and proposed system

## VI. METHODOLOGY

**Firefly algorithm with fittest job first:**

1. Generate initial population of fireflies X randomly.
2. Compute the brightness of each firefly by using objective function
3. Set light absorption coefficient γ
4. While (iteration i<=max)
   Move firefly i to firefly j
   A) Evaluate new fireflies and update brightness then find the machine that has less jobs
   B) Discover each path resulting in the solution.
   C) Apply the path cost mechanism to determine minimum cost path
   D) Apply credit and move to next step
5. Rank fireflies according to their fitness and find the best one.
   If stopping criteria is reached, then go to step 7
6. Else
   go to step-4.
7. Stop.




## VII. CONCLUSION

This paper gives a wide survey of various sorts of metaheuristic strategies which are utilized by scientist to get close ideal arrangements of four unique sorts of jobs in particular BSP, Workflow and DAG, Independent assignments and group-of-Tasks. The different issues and issues found in the individual metaheuristics can be overwhelmed by utilizing the hybridization of at least two metaheuristic procedures. In this paper, an endeavor has been made to give the review and comparison of various metaheuristic algorithms which are utilized to schedule singular kind of jobs.

## REFERENCES


[1] F. Sabahi, "Cloud Computing Security Threats and Responses," pp. 245–249, 2011.

[2] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, and A. Y. Zomaya, "Energy-efficient data replication in cloud computing datacenters," *Cluster Comput.*, vol. 18, no. 1, pp. 385–402, 2015.

[3] N. M. Dhanya and G. Kousalya, "Adaptive and Secure Application Partitioning for Of fl oading in Mobile Cloud Computing," vol. 1, pp. 45–53, 2015.

[4] and M. R. L. Zhang, Z. Zheng, "BFTCloud: A byzantine fault tolerance framework for voluntary-resource cloud computing," *Cloud Comput. (CLOUD), 2011 IEEE Int. Conf.*, pp. 444–451, 2011.

[5] S. Pandey, L. Wu, S. M. Guru, and R. Buyya, "A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments," 2010.

[6] A. Kaur, "A Review on Various Job Scheduling Algorithms," vol. 13, no. 3, pp. 359–367, 2017.

[7] S. Id, W. Count, and C. E. R. Count, "A GA Based Job Scheduling strategy for Computational Grid by Krishan Veer," pp. 29–34, 2015.

[8] R. Baldoni, J. M. Hélary, A. Mostefaoui, and M. Raynal, "On modeling consistent checkpoints and the domino effect in distributed systems," *Rapp. Rech. Natl. Rech. En Inform. En Autom.*, 1995.

[9] X. Zhou and L. Y. He, "A virtualized hybrid distributed file system," *Proc. - 2013 Int. Conf. Cyber-Enabled Distrib. Comput. Knowl. Discov. CyberC 2013*, pp. 202–205, 2013.

[10] M. Arioua, Y. Assari, I. Ez-zazi, and A. Oualkadi, "Multi-hop cluster based routing approach for wireless sensor networks," *Procedia - Procedia Comput. Sci.*, vol. 83, no. Ant, pp. 584–591, 2016.

[11] S. P. Dandamudi, "Parallel Job Scheduling on Multicluster Computing Systems," 2003.

[12] P. Switalski and F. Seredynski, "Scheduling parallel batch jobs in grids with evolutionary metaheuristics," *J. Sched.*, vol. 18, no. 4, pp. 345–357, 2014.

[13] "Content Checked For Plagiarism : al lS eo oo lS."

[14] S. Tahilyani, "A New Genetic Algorithm Based Lane-By-Pass Approach for Smooth Traffic Flow on Road Networks," vol. 1, no. 3, pp. 32–36, 2012.

[15] C. R. Reeves, "A genetic algorithm for flowshop sequencing," *Comput. Oper. Res.*, vol. 22, no. 1, pp. 5–13, Jan. 1995.

[16] W. Wen, C. Wang, D. Wu, and Y. Xie, "An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment," 2015.

[17] B. Li, W. Xu, S. Member, and H. Zhang, "PAPR Reduction for Hybrid ACO-OFDM Aided IM / DD Optical Wireless Vehicular Communications," vol. 9545, no. c, 2017.

[18] A. Hossein and G. X. Yang, "Bat algorithm for constrained optimization tasks," pp. 1239–1255, 2013.

[19] S. Mirjalili and S. Mohammad, "Binary bat algorithm," 2013.

[20] J. Xie, Y. Zhou, and H. Chen, "A novel bat algorithm based on differential operator and Levy flights trajectory," *Comput. Intell. Neurosci.*, vol. 2013, 2013.

[21] I. Engineering, "OPTIMAL POWER FLOW USING CUCKOO OPTIMIZATION ALGORITHM," pp. 4213–4218, 2013.

[22] N. Optimisation, "A comprehensive review of cuckoo search : variants and hybrids Iztok Fister Jr .*, Dušan Fister and Iztok Fister," vol. 4, no. 4, 2013.

[23] K. C. Udaiyakumar and M. Chandrasekaran, "Application of Firefly Algorithm in Job Shop Scheduling Problem for Minimization of Makespan," *Procedia Eng.*, vol. 97, pp. 1798–1807, 2014.

[24] B. Jiang, J. Wu, X. Zhu, and D. Hu, "Priority-Based Live Migration of Virtual Machine," pp. 376–385, 2013.

[25] M. R. Abid, K. Kaddouri, K. Smith, M. Idriss, E. Ouadghiri, and M. Gerndt, "Virtual Machines ' Load-Balancing in Inter-Clouds," pp. 5–12, 2016.

[26] X. Y. S. Deb, "Cuckoo search : recent advances and applications," 2013.

[27] X. Cui, B. Mills, T. Znati, and R. Melhem, "Shadow replication: An energy-aware, fault-tolerant computational model for green cloud computing," *Energies*, vol. 7, no. 8, pp. 5151–5176, 2014.

[28] A. Thomas, G. Krishnalal, and J. R. V P, "Credit Based Scheduling Algorithm in Cloud Computing Environment," *Procedia - Procedia Comput. Sci.*, vol. 46, no. Icict 2014, pp. 913–920, 2015.

[29] N. Kord and H. Haghighi, "An Energy-Efficient Approach for Virtual Machine Placement in Cloud Based Data Centers," pp. 44–49, 2013.

[30] J. Li, C. Pu, Y. Chen, V. Talwar, and D. Milojicic, "Improving Preemptive Scheduling with Application-Transparent Checkpointing in Shared Clusters," pp. 222–234.