*Research Article*

# Performance Analysis of Computational Grid Job Scheduling using Bio-Inspired Heuristic Function

**Kishor Shamrao Sakure[1], Rajesh Kumar Boghey[2]**

Department of Computer Science & Engineering[1,2]
Madhyanchal Professional University, Bhopal, India[1,2]
kishor_sakure @yahoo.com[1], rajeshboghey@gmail.com [2]

**Abstract**
The increasing rate of jobs and a limited number of resources decline the performance of the computational grid. Therefore, the scheduling of tasks plays a vital role in the computational grid. The conventional scheduling of the computational grid applies the CPU scheduling algorithms such as FCFS, SJF and round-robin. However, the limited constraints factors of scheduling algorithms increase the ratio of job failure and degrade the overall performance of computational grids. Therefore, the incremental research approach uses a bio-inspired heuristic function to focus on the task scheduling algorithm. Furthermore, the searching capacity of the bio-inspired function increases the utilization of resources such as CPU and memory to share resources. This paper presents the experimental analysis of various algorithms such as ACO, PSO, GSO, ABC and TLBO to schedule tasks in a computational grid of different sizes such as 10 X 10, 20X20 and 40 x 40. the simulation software uses MATLAB version R2014a. The empirical evaluation of the systems is estimated with job failure and job completion.

**Keywords: -** Task Scheduling, Swarm Intelligence, Computational Grids · Jobs Failure, Resource Utilization

## I. Introduction

Grid computing is a subgroup of distributed computing model. Distributed computing focuses on resources such as CPU, memory, network bandwidth and storage space. The power of grid computing is handling the large project of multiple applications worldwide. Grids are classified into several types, including data-grids, utility grids, smart grids, etc. This type of distributed computing has many applications, including weather forecasting, monitoring earthquakes and volcanic eruptions, researching new medicines, and so on. Because the grid's heterogeneous resources are dispersed across a large geographical area, they must be exploited efficiently. The goal is to maximize the use of underutilized resources. As a result, given that resource allocation in grids is an NP-Complete problem, it must be addressed effectively. According to a literature review, there is no prescribed optimal solution or final solution to the NP-complete problem [1,2,3]. It is well known that the job scheduling problem is NP-complete [4]. Genetic algorithms were recently introduced in application-level scheduling to minimize the average completion time of jobs through optimal job allocation on each grid node [5, 6]. Because of the intractable nature of the problem and its importance in grid computing, it is preferable to investigate alternative approaches to developing good heuristic algorithms for the problem. The grid resource broker is in charge of resource discovery, assigning jobs to specific grid nodes, binding user applications (files), hardware resources, initiating computations, adapting to changes in grid resources, and presenting the grid to the user as a single, unified resource [11]. Finally, it manages the available resources and controls the physical allocation of tasks while dynamically updating the grid scheduler whenever there is a change in resource availability. Knowing the processing speeds of available grid nodes and the job length of user applications is a time-consuming task in a grid environment. It is usually simple to obtain information about the speed of available grid nodes, but it is much more challenging to determine the computational processing time. Only dynamic scheduling will be helpful when the computing power demand exceeds the available grid nodes. To think of the problem as an algorithm, we must dynamically estimate job lengths based on user application specifications or historical data[8.9]. To achieve better performance of the Computational grid, the resources must be utilized at the maximum potential to reduce their idle time. The objective of grid scheduling is to utilize the available resources in the distributed environment. The grid scheduler collects resource status information, selects appropriate resources, and determines the best schedule for executing applications on a Grid system. Because of the grid environment's dynamic nature, developing scheduling algorithms is always a major challenge for researchers. The scheduling must be done in a way to maximize resource utilization. In the Grid system, scheduling decisions must be made in the shortest makespan, users competing for resource allocation, and time slots desired by users could be dynamic. The goal of any task scheduling is to achieve minimum makespan and relevant resources for each Job to satisfy the user's constraints. The Grid scheduling problem has been taken to be an NP-hard optimization problem—this paper study the process of computational grid computing job allocation and resource allocation process. For the allocation of Jobs and resources, there is no standard method for executing the task. The computational grid used the first-come, first allocation technique (FCFS) in the average case. In the case of the FCFS technique, the failure of Job is very high, and in all cases, the performance of grid computing is degraded. Now a day's, various authors used a heuristic and meta-heuristic function for the improvement of the computational grid. Consequently, these functions, such as genetic algorithm, calming hill algorithm, and various algorithms, are used. Some authors focus on the ant colony optimization technique for allocating jobs[14,15]. If the job size is increasing, the allocation of Jobs has faced a problem of the selection method. The process of selection method used different constraints function for the selection of Job. The rest of paper organized as in section II Related Work. In section III Bio-inspired Function, in section IV experimental analysis of Grid, and in section V discuss the conclusion and future work.

**II. Related Work**

The continuous efforts of algorithm development in the area of swarm intelligence, increase the capacity of computational grid. The allocation of resource is major challenge in distributed manners in complex project structure. Some common contribution of algorithm describe here.

In this [1] author proposed feature-based local (group) and global (network) formation procedures are combined with Internet of Things (IoT)-based solutions to provide the best possible outcome. the simulation-optimization constraints integrated model improves route identification for small-scale IoT networks by a minimum of 3.4 percent and a maximum of 16.2 percent.

In this [2] author give a comprehensive review of the literature from various perspectives, including active researchers, research motivations, QoS parameters, algorithms, datasets, optimization strategies, and possible layers to identify areas that require more research. The findings show that the most important motive for researchers is to reduce response time, and meta-heuristic algorithms, particularly genetic algorithms, are the most extensively employed computational intelligence techniques.

In this [3] author proposed MAOPSO particle swarm optimization technique is addressed in terms of four competing objectives: dependability maximization, cost minimization, make pan, and energy consumption. When compared to the LEAF, MAOPSO, and EMS-C algorithms, outcome demonstrate that the presented approach can improve the Hyper Volume (HV) criterion by up to 71 percent, 182 percent, and 262 percent, respectively.

In this [4] author proposed the optimization is done via the Hybrid Adaptive Particle Swarm Optimization (HAPSO) algorithm, which is a hybrid of Genetic Algorithm and Adaptive Particle Swarm Optimization. With a mean square error reduction of 98.92 percent, HAPSO converges quicker than Standard Particle Swarm Optimization and Self Adaptive Particle Swarm Optimization.

In this [5] author proposed CTDHH approach was evaluated in a real-world cloud-based experimentation environment by comparing it to five baseline approaches, namely four population-based approaches and an existing hyper-heuristic approach called Hyper-Heuristic Scheduling Algorithm (HHSA).

In this [6] author proposed approach is used to maximize the suggested service composition in terms of aggregating various QoS criteria as fitness values. A privacy-aware cloud service composition approach with respect to QoS optimization in the IoT environment is discussed by presenting an IoT-based cloud service composition conceptual model for the privacy level computing model, as well as a novel hybrid evolutionary algorithm called SFLA-GA that combines the SFLA and GA.

In this [7] author analyzing the characteristics of DEPMSP yields five qualities. KTPO is compared against four algorithms from the literature in extensive simulation trials. The comparative results and statistical analysis show that KTPO is successful and advantageous in the treatment of DEPMSP.

In this [8] author examines technologies, procedures, and strategies that can be utilized to increase the reliability of distributed applications in diverse and heterogeneous network settings, as well as the problem of dependable resource provisioning in joint edge-cloud systems. The outcome of the survey is presented, as well as a problem-oriented assessment of the current state of the art.

In this [9] author proposed The Cloud Sim simulator tool is used to compare the implementation of their discussed system to that of the benchmarked schemes. Simulation results show that their discussed DMOOTC scheme provides better service choices with minimum total execution time and cost for the first scenario, and achieves 21.64 percent, 18.97 percent, and 13.19 percent improvement in terms of execution time for the second scenario compared to that of the first.

In this [10] author present the comprehensive review and classification of the discussed scheduling approaches, as well as their benefits and drawbacks. They anticipate that their thorough and comprehensive survey effort will serve as a stepping stone for young cloud computing researchers and will aid in the development of scheduling techniques.

In this [11] author proposed a new resource provisioning technique to support data-intensive applications' deadline requirements in hybrid cloud settings. The outcome of an experiment using a real case study executing a data-intensive application to measure the walkability index on a hybrid cloud platform made up of dynamic resources.

In this [12] author proposed Cloudlet and public clouds are used in the presented technique to provide a more energy-efficient offloading strategy for home automation applications. To schedule mobile services, a particle swarm optimization (PSO)-based heuristic technique is employed. Extensive tests are carried out to demonstrate the effectiveness and efficiency of the algorithm in question.

In this [13] author provides a comprehensive overview of load balancing techniques. The merits and drawbacks of existing systems are discussed, as well as key problems that must be overcome in order to build efficient load balancing algorithms. The research also offers new perspectives on load balancing in cloud computing.

In this [14] author proposed a multi-factor evaluation-based intelligent GBMFPA model is examined to improve production allocation. For the sale of rare and antique items, the bid-market model is built with an intermediate well-informed arrangement between the grid server and the clients. Dynamic Price Updating and Incentive-and-Profile Driven Scheduling are two types of architectural processing.

In this [15] author proposed a CEFA for resolving workflow scheduling issues in an IaaS platform The CEFA under discussion employs a novel method for problem encoding, population initialization, and fitness evaluation with the goal of delivering cost-effective and optimal workflow execution within a specified time frame. Their findings show that the described CEFA beats current state-of-the-art heuristics in terms of meeting the deadline constraint and minimizing the cost of execution.

In this [16] author offer an adaptive fault-tolerant scheduling system that combines dynamic replication and rescheduling. The system first schedules jobs based on the completion time and fault rate of resources, and then applies fault-tolerant techniques based on resource availability. Experiments were conducted to assess performance, and it was discovered that the suggested scheduling system surpasses the present system by a factor of 4.8 percent Average task reaction time and 0.02 percent Average flowtime.

In this [17] author presents a new hybrid feature selection strategy combining a GA with a SVM. To optimize parameters and identify appropriate feature subsets, the GA–SVM model is used. The GA-5-fold SVM's cross-validation accuracy is used to assess the defect detection capabilities of feature subsets, and then a novel subset is chosen as the best feature subset.

In this [18] author analysis various load balancing algorithms and brokering policies used in load balancers for each service and its scheduling kinds by methodically analyzing recent approaches established for load balancing and service brokering (the most significant challenge), the survey's objectives are to determine, demonstrate, compare, and assess them. Classify and assess cloud computing techniques based on the essential parameters.

In this [19] author proposed the importance of optimization methods and the advantages they offer in overcoming cloud load-balancing difficulties Furthermore, in order to handle the load balancing problem in cloud environments, the advantages and disadvantages of nature inspired meta-heuristic algorithms have been explored, as well as their main problems, in order to provide more effective strategies.

In this [20] author provide a new resource availability characterization and prediction method for dynamic heterogeneous distributed systems to achieve this goal. The use of three data mining technologies especially the neural network to model and predict resource availability based on their discovered availability attributes is explored. For instant and duration availability, the predictions made using the given approach are 18% and 31% more accurate on average than those made using the best method (Naive Bayes' Classifier).

In this [21] author presented smart grid dynamic regulating pools, a system and method. At least one processor in the system may be set to start a number of pool regulation jobs, each of which is handled by a different processing resource.

In this [22] author proposed a novel energy-efficient container-based scheduling (EECS) technique for quickly processing various sorts of IoT and non-IoT workloads The presented method employs the accelerated particle swarm optimization (APSO) technique to quickly select the best container for each task. the EECS technique can deploy the containers on the best cloud server possible, using the best scheduling strategy possible.

In this [23] author demonstrates that the DCLCA outperformed the MTCT, MAXMIN, ant colony optimization, and genetic algorithm-based NSGA-II in terms of lower make span, with improvements of 57.8, 53.6, 24.3, and 13.4 percent in the first scenario and 60.0, 38.9, 31.5, and 31.2 percent in the second scenario.

In this [24] author proposed PSO and GA have different effects on workflow scheduling. To assess the metaheuristics' performance, a security and cost-aware workflow scheduling algorithm was chosen. With a risk rate constraint ranging from 0 to 1 with a 0.1 step, three methods were assessed in three real-world workflows. According to the findings, GA-based algorithms outperformed the PSO in terms of cost-effectiveness and response time.

In this [25] author provides an application-aware job scheduling mechanism (AJSM) that includes a heuristic application-aware deadline constraint job scheduling algorithm and a periodic scheduling flow. The results of the rigorous performance evaluation clearly show that the described application-aware task scheduling mechanism can schedule more Grid jobs successfully than the previous algorithms.

In this [26] author address the mobility needs of hybrid networks; better particle swarm optimization is integrated into the quality service evaluation of dynamic service composition. This research divides hybrid services into different task groups in order to generate candidate sets, and then uses interface matching to compare the operations of candidate services to user needs in order to choose the best service.

In this [27] author proposed a new cloud manufacturing multi-task scheduling model based on game theory. The Nash equilibrium point in the game determines the best outcome for a cloud manufacturing platform. Because the cloud manufacturing multi-task scheduling problem is an NP-hard combinatorial optimization problem, they provide an enhanced biogeography-based optimization algorithm that incorporates three improvements to solve the model.

In this [28] author conducted a comprehensive literature review of state-of-the-art cloud load balancing techniques. Traditional approaches, heuristics, meta-heuristics, and a hybrid approach are all part of the algorithm. This work conducts a thorough historical review and analysis of the existing load balancing (LB) literature. Researchers will be able to use the work provided here to assist them create new efficient load balancing algorithms in the Cloud computing area.

In this [29] author proposed the state-of-the-art approaches in fuzzy logic-based cloud computing and their key aspects They begin by providing an overview of cloud computing and a categorization of current research projects. Second, they provide a synopsis of related research works as well as some of the major methodologies described in current literature. Finally, they make some suggestions for further research in the topic.

In this [30] author addressed value-based optimization, which may be applied to a wide range of production processes. Value is defined in terms of OEE, which is measured in monetary units. They talked about a new online halting condition that considers the expected utility of additional computing effort. They compare this strategy to a baseline stopping criterion with no prediction mechanism for solving scheduling problems in the algebra. The described approach yields near-optimal profit in a variety of issue scenarios.

## III. Bio-Inspired Function

The conventional scheduling algorithms have several bottleneck problems during the processing of jobs in a computational grid system. Therefore, most authors applied conventional FCFS, SJF, and round-robin methods. These conventional methods are outstanding results in case of limited jobs and resources. However, the allocation process in distributed manners faces a problem of job failure and maximum period in the job queue. the Bio-inspired function removes the bottleneck problems from the computational grids. It enhances the process of jobs allocation and completion time[15]. The swarm intelligence has a heuristic search capacity, and ample space for job mapping reduces the job failure situation in computational grids. The major swarm intelligence algorithms such as Ant colony optimization (ACO), particle swarm optimization (PSO), glowworm swarm optimization, ABC algorithm and TLBO algorithm. The processing of algorithm in computational grids describe here.

### Ant colony optimization (ACO)

The motivation behind the method is to adjust the memory and to diminish the make-traverse. In a subterranean insect framework, an insect is represented as a vocation in a matrix domain. Pheromone esteem on a way in insect framework is identical to the heaviness of an asset in lattice framework. The computational limit of an asset is increasingly when the heaviness of asset is expansive. The scheduler gathers the client information and processes the heaviness of assets accessible. The heaviness of asset is utilized as a parameter by the scheduler for the booking method. At that point scheduler chooses an asset and executes the occupation presented by client and finally presents the outcomes acquired to the client. At first, the pheromone estimation of every asset for every occupation is equivalent to the pheromone pointer (PV). For every employment the estimation of PV of every asset is processed by including the execution time and expected transmission time of the occupation submitted. The normal time can be effectively controlled by Sj and Bandwidthi . Here Sj is the span of a given employment j and bandwidthi is the asset and the scheduler[16].

$$PV_{IJ} = [\ \frac{SJ}{BWi} + \frac{Tj}{speed\ X\ (1 - Li)}]^{-1}$$

Get number of resources from num_resource

Get number of jobs from num_gridlet

Compute pheromone indicator value;

For each num_resource:

Bandwidth = getBandwidth();

MIPS = getMIPS();

load = getLoad();

createResource(id, bandwidth, MIPS, load);

For each num_gridlet:

job_size = getJobSize();

cpu_time = getCPUTime();

createGridlet ();

createAnt();

Calculate PV for each resource and job combination

Store the value in PV Matrix

While process_iteration<num_gridlet

For (R = 0; R <num_resource; R++)

**Particle swarm optimization (PSO)**

The dynamic job allocation process accrued the process of scheduling based on task and resource allocation. For the allocation of task and resource used two different technique one is searching of task according to the dedicated job and other is execution of task incorporation of process. For the execution of task used particle of swarm optimization. The particle swarm optimization the load and perform the task[18].

The following parameter is used for the process of job allocation methods, x1, x2,…………..,xn is the loads component of user side over cloud network. W is the Wight factor for the sum of loads, $\tau$ is the value of pheromones of ants, v1 and v2 is velocity of particle agents, c1 and c2 is constants value of particle. The process of selection step given below.

Step1. Define the value of loads set S1{x1,x2,………….xn} with population random population of PSO.

a. Assign the velocity of particle V1=0,V=0 and W=0
b. Fitness constrains function for the selection of ants

$$F(s) = \frac{(Ffd - Fpf)}{Fd * fp}, wi \epsilon S(x1, x2 \ldots\ldots xn) \ldots\ldots\ldots\ldots\ldots (1)$$

Here Ffd is assigned resource and Fpf is allocated and w is set of resource mapping component of sum sets

The selected loads components set the value of ants$F = \{fa1 \ldots\ldots\ldots an\}$. these ants value proceed for the estimation of local best, the local best function define as

$$Pbest = \begin{cases} \dfrac{(\tau_i)^{\alpha}\left(LI_i^{S_j}\right)^{\beta}}{\sum_{g \notin S_j}(\tau_g)^{\alpha}\left(LI_g^{S_j}\right)^{\beta}} & if \quad i \notin S_j \\ 0 & otherwise \end{cases} \qquad (2)$$

Here $\tau_i$ is phenomenon value of ants and LI is value of least interface of ants.

Step2. The Pbest value set to Gbest

Input the job allocation state of Gbest Value

1. Calculate the value of relative mapping set in Gbest set
   $\mathrm{Rf} = \frac{\mathrm{LSI}}{\mathrm{Wd}}$ Here Lsi is interference value of ants and Wd is sum value of PSO space.
2. The PSO space creates the selection state for the processing of mapping load.
   $$FS = \begin{cases} \dfrac{\max\ (RF)\ - F(s)}{\max_{h=1:}(WS)} & if\ \ s_i\ \in f_j \\ 0 \quad otherwise \end{cases} \qquad (3)$$
3. create the relative FS difference value

$$\mathrm{Rd} = \sum_{\mathrm{fd}=1}^{\mathrm{n}} \sum_{\mathrm{pf}=1}^{\mathrm{m}} (\mathrm{xi} - \mathrm{fs}) \ldots \ldots \ldots \ldots \ldots \ldots (4)$$

4. if the value of Rd is zero the job allocation process is done.
5. Else the process of selection goes into steps 2
6. if loads selection state is empty the process of selection is terminated.

Glowworm swarm optimization (GSO)

The glow-worm optimization algorithm is new swarm intelligence algorithm without the processing of memory. The process of algorithm used the concept of local sharing information and update the jobs set of parameters.

The process of algorithms defines following parameters

1. The jobs of grid data encode i with objective function J(xi(t)) the position of jobs data is xi(t) into acceleration α. The value of luciferin li spread with neighbour (Ni(t)) of glow-worm. Each iteration of jobs set the new factor of decision is updated by equation (1)

$$r_d^i(t+1) = min\left\{rs, max\left\{0, r_d^i(t) + \beta(nt - |Ni(t)|)\right\}\right\} \ldots \ldots \ldots \ldots (1)$$

After the iteration of jobs of new neighbours is

$$N_{i(t)=\left\{j:\|xi(t)-xi(t)\|<r_d^i;li(t)<li(t)\right\}} \ldots \ldots (2)$$

The movement of jobs component by local decision is.

$$p_{ij(t)=\frac{l_{i(t)}-l_{i(t)}}{\sum_{k\in Ni(t)} lk(t)-li(t)}} \ldots \ldots (3)$$

Update the new set of jobss

$$x_{i(t+1)=xi(t)+s\left(\frac{sj(t)-xi(t)}{\|xj(t)-xi(t)\|}\right)} \ldots \ldots (4)$$

Update the value if luciferin

$$l_{i(t)=(1-\rho)li(t-1)+\gamma j(xi(t))} \ldots \ldots (5)$$

Any finally job is allocated to computational grid.

Teacher learning based optimization (TLOB)

This enhancement strategy depends on the impact of the impact of an educator on the yield of learners in a class. It is a populace-based technique and like other populace-based strategies it utilizes a populace of answers for continue to the worldwide arrangement. A gathering of learners constitutes the populace in TLBO [41]. In any streamlining calculations there are quantities of various outline factors. The diverse outline factors in TLBO are comparable to various subjects offered to learners and the learners' outcome is undifferentiated from the 'wellness', as in other populace-based enhancement methods. As the instructor is viewed as the most learned individual in the general public, the best arrangement so far is similar to Teacher in TLBO. The procedure of TLBO is partitioned into two sections. The initial segment comprises of the "Instructor stage" and the second part comprises of the "Learner stage". The "Instructor stage" implies gaining from the educator and the "Learner stage" implies learning through the communication between learners. In the sub-area’s underneath, we quickly talk about the execution of TLBO.

Initialization

Following are the notations used for describing the TLBO

N: number of learners in class i.e. “class size”

D: number of courses offered to the learners

MAXIT: maximum number of allowable iterations

The population X is randomly initialized by a search space bounded by matrix of N rows and D columns. The jth parameter of the ith learner is assigned values randomly using the equation

$$x^{0}_{(i,j)=x_j^{min}} \; +rand \times \left(x_j^{max} - x_j^{min}\right) \text{.............................}(1)$$

where rand represents a uniformly distributed random variable within the range (0, 1), xmin j and xmaxj represent the minimum and maximum value for jth parameter. The parameters of ith learner for the generation g are given by

$$X^{g}_{(i)} \; = \left[x^{g}_{(i,1)}, x^{g}_{(i,2)}, \dots\dots\dots\dots.\, x^{g}_{(i,j)\text{..........}, x^{g}_{(i,D)'}}\right] \dots\dots\dots\dots\dots..(2)$$

### III.1 Teacher phase

The mean parameter Mg of each subject of the learners in the class at generation g is given as

$$M^{g} = \left[m^{g}_{1}, m^{g}_{\;2,\text{..........................}m^{g,\text{....................}m^{g}_{D}}_{j}}\right] \dots\dots\dots\dots\dots\dots\dots\dots\dots.(3)$$

The learner with the base target work esteem is considered as the instructor Xg Teacher for particular cycle. The Teacher stage makes the calculation continue by moving the mean of the learners towards its instructor. To get another arrangement of enhanced learners an arbitrary weighted differential vector is framed from the present mean and the wanted mean parameters and added to the current populace of learners.

$$Xnew^{g}_{(i)=X^{g}_{(i)+rand \;\times\left(X^{g}_{Teac\,her} - TFM^{\,g}\right)}} \text{..........................................}(4)$$

TF is the teaching factor which decides the value of mean to be changed. Value of TF can be either 1 or 2. The value of TF is decided randomly with equal probability as,

$$T_{F=round\;[1+rand\;(o,1)\{2--1\}]} \dots\dots\dots\dots\dots\dots.(5)$$

Where TF is not a parameter of the TLBO calculation. The estimation of TF is not given as a contribution to the calculation and its esteem is haphazardly chosen by the calculation utilizing Eq. (5). In the wake of leading various investigations on numerous benchmark capacities it is inferred that the calculation performs better if the estimation of TF is somewhere around 1 and 2. Notwithstanding, the calculation is found to perform much better if the estimation of TF is either 1 or 2 and thus to improve the calculation, the showing variable is recommended to take either 1 or 2 relying upon the gathering together criteria given by Eq. (5). If Xnew is found to be a superior learner than Xg in generation g , than it replaces inferior learner Xg in the matrix.

### III.2 Learner phase

In this stage the association of learners with each other happens. The procedure of common cooperation tends to build the information of the learner. The arbitrary connection among learners enhances his or her insight. For a given learner Xg , another learner Xg is randomly selected (i≠ r). The ith parameter of the matrix Xnew in the learner phase is given as

$$Xnew^{g}_{(i)=\{x^{g}_{i} + rand\,\times\left(X^{g}_{i} - X^{g}_{r}\right)\; if\; f\left(X^{g}_{i}\right) < f\left(X^{g}_{r}\right)}$$
$$X^{g}_{i} + rand \times \left(X^{g}_{r} - X^{g}_{i}\right)\; otherwise \;\text{..................}(6)$$

### III.3 Algorithm termination

The algorithm is terminated after MAXIT iterations are completed.

## IV. Experimental Analysis

To evaluate the performance of bio-inspired function for computational grid simulation in MATLAB programming language. The process of simulation design different sizes of grids such as 10 X10, 20X20 and 40X 40. The performance parameters were measured as job completion and failure rate[25,27,28,31].

### Job completion

Job completion is one of the most important standard metrics used to measure the performance of fault tolerant systems [5]. Job completion is defined as:

$$job\ Completion = \frac{n}{T}$$

Where $n$ is the total number of jobs submitted and $Tn$is the total amount of time necessary to complete $n$ jobs. Job completion is used to measure the ability of the grid to accommodate jobs.

**Job Failure**

It is the percentage of the tendency of the selected grid resources to fail and is defined as:

$$failure = \frac{\sum_{j=1}^{m} Pfj}{m} \times 100\%$$

Where $m$ is the total number of grid resources and $Pfj$is the failure rate of resource $j$. Through this metric, the faulty behavior of the system can be expected.

**Table 1: Job Failure and Job Completion Rate of small job for MC-ACO gird size 10 X10**

| **Small Job (MC-ACO)** | | | |
|---|---|---|---|
| **No. of Jobs** | **No. of Resources** | **Job Failure Rate** | **Job Completion Rate** |
| 100 | 50 | 6.168 | 99.167 |
| 200 | 100 | 5.002 | 86.568 |
| 300 | 150 | 5.003 | 86.835 |
| 400 | 200 | 4.006 | 84.965 |
| 500 | 250 | 5.008 | 85.455 |
| 600 | 300 | 5.334 | 88.335 |
| 700 | 350 | 4.552 | 84.389 |
| 800 | 400 | 4.001 | 83.535 |
| 900 | 450 | 4.555 | 82.335 |
| 1000 | 500 | 4.568 | 83.333 |

**Table 2: Job Failure and Job Completion Rate of small job for MC-PSO grid size 10 X 10**

| **Small Job (MC-PSO)** | | | |
|---|---|---|---|
| **No. of Jobs** | **No. of Resources** | **Job Failure Rate** | **Job Completion Rate** |
| 100 | 50 | 2.001 | 91.168 |
| 200 | 100 | 4.503 | 95.458 |
| 300 | 150 | 4.335 | 96.179 |
| 400 | 200 | 2.008 | 91.525 |
| 500 | 250 | 3.009 | 89.665 |
| 600 | 300 | 6.965 | 96.525 |
| 700 | 350 | 3.008 | 86.169 |
| 800 | 400 | 3.965 | 86.555 |
| 900 | 450 | 3.864 | 90.333 |
| 1000 | 500 | 3.758 | 87.598 |

**Table 3: Job Failure and Job Completion Rate of Middle job for MC-GSO grid size 20 X 20.**

| **Middle Job (MC-GSO)** | | | |
|---|---|---|---|
| **No. of Jobs** | **No. of Resources** | **Job Failure Rate** | **Job Completion Rate** |
| 1000 | 500 | 4.005 | 84.168 |
| 2000 | 1000 | 4.555 | 83.333 |
| 3000 | 1500 | 4.595 | 82.355 |
| 4000 | 2000 | 4.255 | 83.555 |
| 5000 | 2500 | 5.333 | 88.365 |
| 6000 | 3000 | 4.565 | 84.385 |
| 7000 | 3500 | 4.585 | 86.595 |
| 8000 | 4000 | 7.365 | 86.698 |
| 9000 | 4500 | 6.165 | 89.756 |

| | | | |
|---|---|---|---|
| 10000 | 5000 | 4.955 | 84.595 |

**Table 4: Job Failure and Job Completion Rate of Large job for MC-ACO gird size 20 X 20.**

| **Large Job (MC-ACO)** | | | |
|---|---|---|---|
| **No. of Jobs** | **No. of Resources** | **Job Failure Rate** | **Job Completion Rate** |
| 10000 | 5000 | 4.001 | 84.555 |
| 20000 | 10000 | 4.533 | 83.000 |
| 30000 | 15000 | 4.008 | 83.956 |
| 40000 | 20000 | 4.000 | 84.458 |
| 50000 | 25000 | 4.965 | 82.333 |
| 60000 | 30000 | 3.335 | 87.655 |
| 70000 | 35000 | 4.565 | 83.595 |
| 80000 | 40000 | 4.955 | 84.555 |
| 90000 | 45000 | 4.008 | 85.667 |
| 100000 | 50000 | 4.555 | 84.565 |

**Table 5: Job Failure and Job Completion Rate of Large job for MC-TLBO 20 X20.**

| **Large Job (MC-TLBO)** | | | |
|---|---|---|---|
| **No. of Jobs** | **No. of Resources** | **Job Failure Rate** | **Job Completion Rate** |
| 10000 | 5000 | 3.008 | 90.565 |
| 20000 | 10000 | 3.010 | 89.666 |
| 30000 | 15000 | 3.055 | 88.655 |
| 40000 | 20000 | 3.065 | 91.255 |
| 50000 | 25000 | 3.655 | 86.545 |
| 60000 | 30000 | 2.333 | 95.333 |
| 70000 | 35000 | 3.555 | 89.333 |
| 80000 | 40000 | 3.455 | 87.365 |
| 90000 | 45000 | 3.009 | 92.895 |
| 100000 | 50000 | 3.955 | 94.689 |

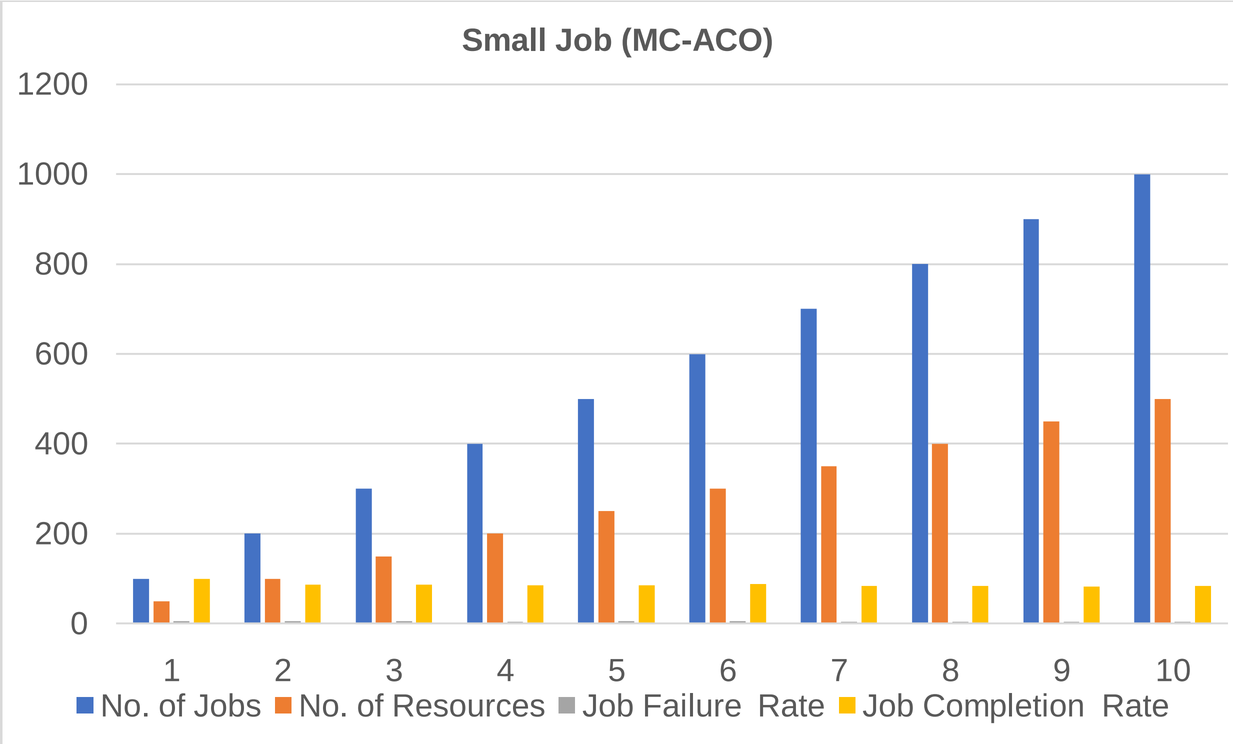


**Fig:1 Comparative result analysis of Small job (MC-ACO) of 10*10 small grid models.**

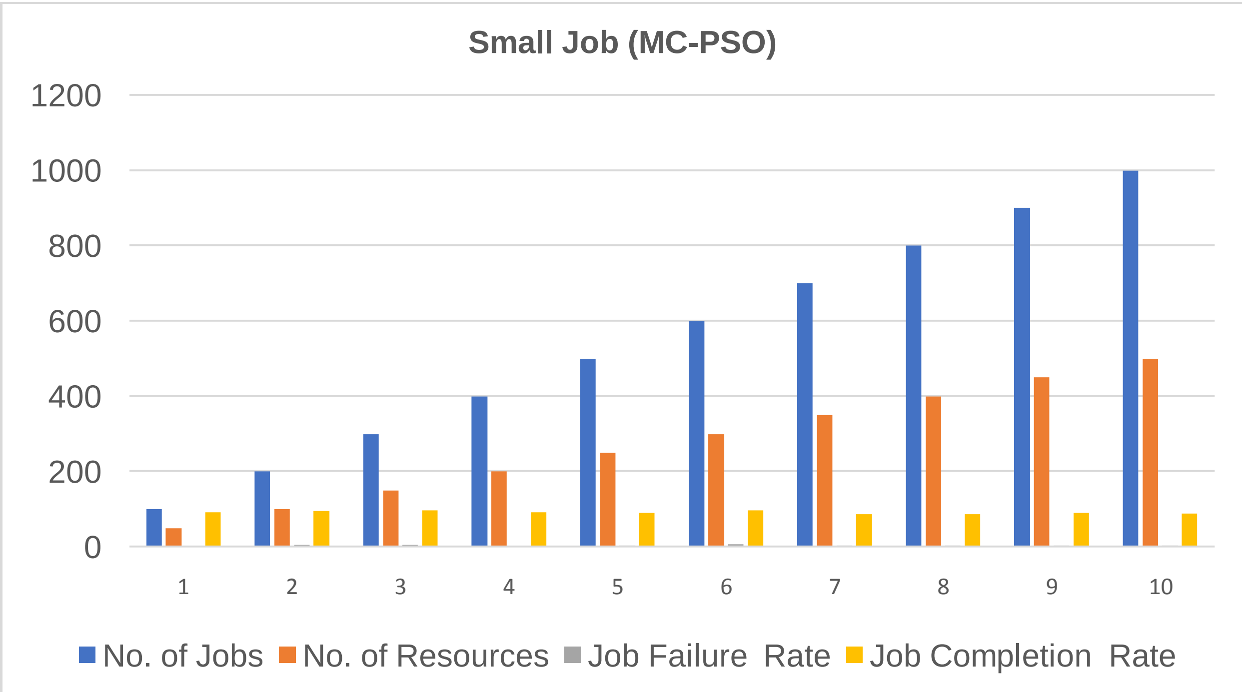


**Fig:2 Comparative result analysis of Small job (MC-PSO) of 10*10 small grid models.**

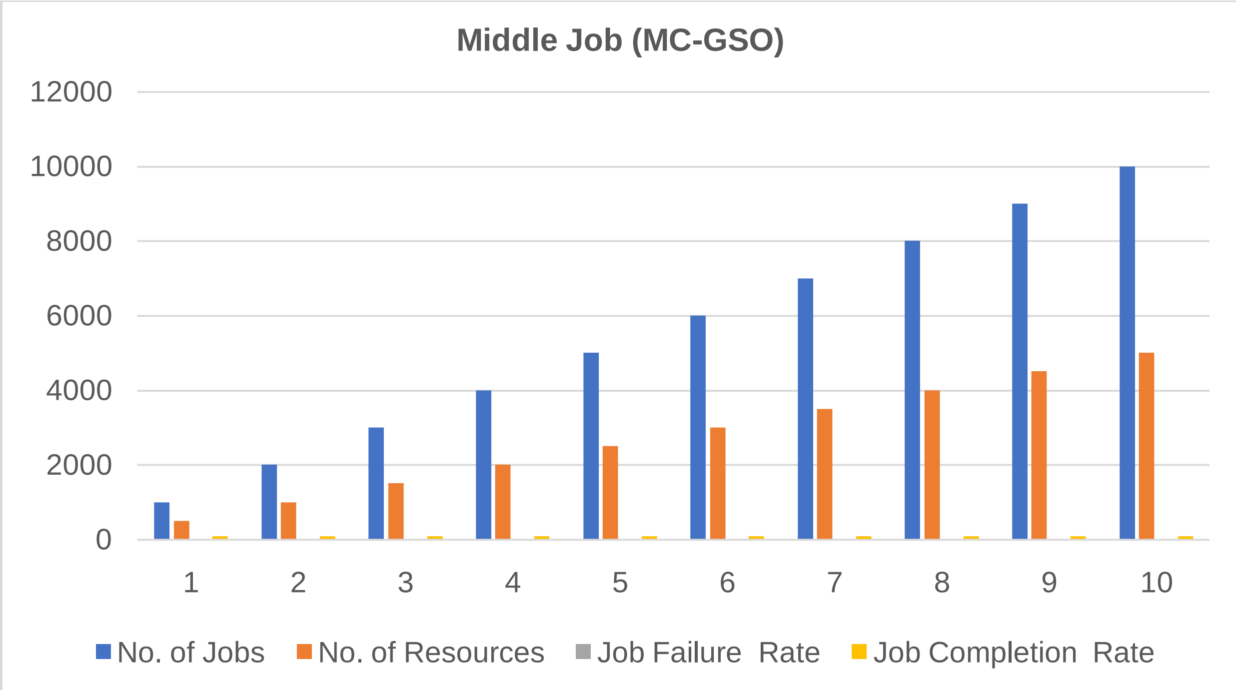


**Fig: 3 Comparative result analysis of middle job (MC-GSO) of 20*20 small grid models.**

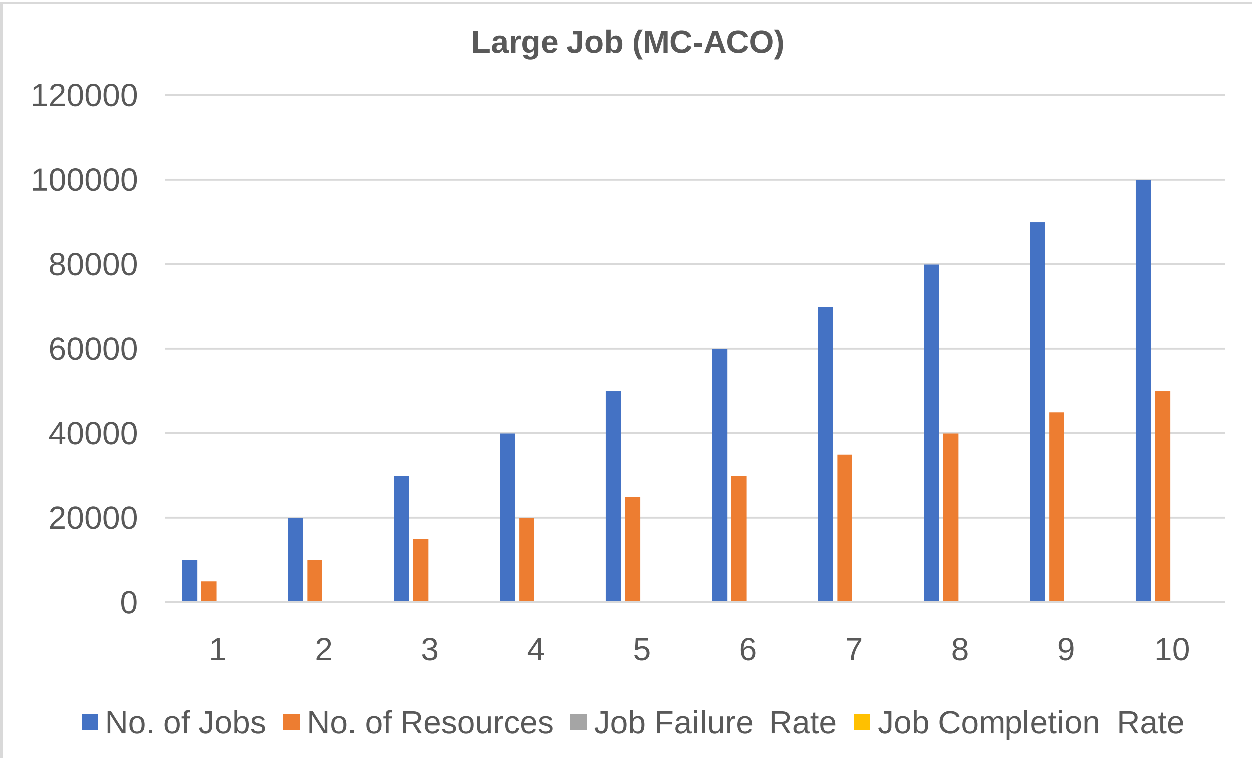


**Fig: 4 Comparative result analysis of large job (MC- ACO) of 40*40 large grid models.**

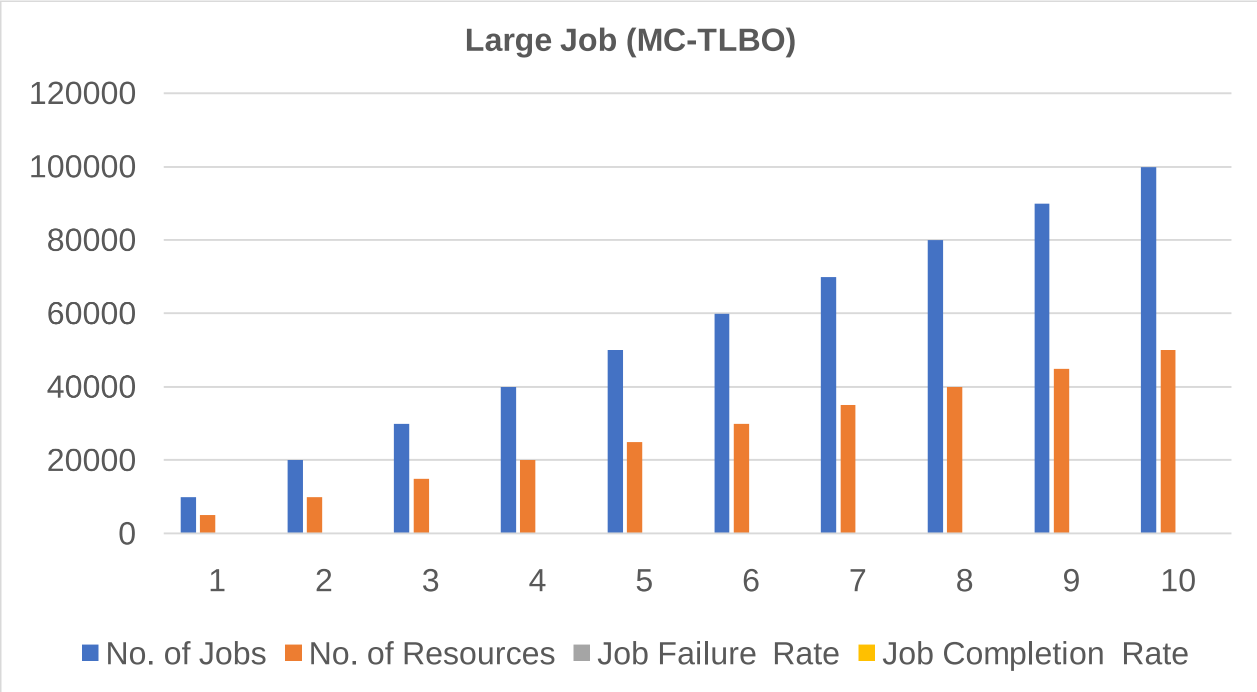


**Fig: 5 Comparative result analysis of Large Job (MC-TLBO) of 40*40 large grid models.**

## V. Conclusion & Future Work

The recent development in computational grid expanding to accommodate the increasing demand of resource for large scale application. The expansion of resource allocation faces a problem of job failure and maximum time span. The swarm intelligence-based algorithm reduces the job failure constraints of expanding the resource allocation process over the geographically location. in this paper, a swarm intelligence-based policy such as PSO, ACO, and TLBO for scheduling parallel jobs consisting of collaborative tasks in multi-location Grid. The TLOB scheduling policy effectively combines the attractiveness feature of the learning algorithm with robust global search particle mechanism of PSO improved searching efficiency and diversity in the population of scheduling solutions over successive generations. The extensive analysis was carried out with three grid sizes. The scheduling policy is evaluated against many heuristics and meta-heuristic scheduling techniques using test cases consisting of different sized workloads, a variable number of resources, and diverse resource heterogeneities. The obtained experimental results have proved the efficacy TLBO algorithm. TLBO policy also offers scalability by producing good results with a scaled number of jobs and resources.

## References


1. Kumar, Adarsh, Krishnamurthi Rajalakshmi, Saurabh Jain, AnandNayyar, and Mohamed Abouhawwash. "A novel heuristic simulation- optimization method for critical infrastructure in smart transportation systems." International Journal of Communication Systems 33, no. 11 (2020): e4397.
2. She, Qiping, Xiaochao Wei, GuihuaNie, and Donglin Chen. "QoS-aware cloud service composition: A systematic mapping study from the perspective of computational intelligence." Expert Systems with Applications 138 (2019): 112804.
3. Saeedi, Sahar, ReihanehKhorsand, SomayeGhandiBidgoli, and MohammadrezaRamezanpour. "Improved many-objective particle swarm optimization algorithm for scientific workflow scheduling in cloud computing." Computers & Industrial Engineering 147 (2020): 106649.
4. Midya, Sadip, Asmita Roy, KoushikMajumder, and SantanuPhadikar. "Multi-objective optimization technique for resource allocation and task scheduling in vehicular cloud architecture: A hybrid adaptive nature inspired approach." Journal of Network and Computer Applications 103 (2018): 58-84.
5. Alkhanak, Ehab Nabiel, and Sai Peck Lee. "A hyper-heuristic cost optimisation approach for Scientific Workflow Scheduling in cloud computing." Future Generation Computer Systems 86 (2018): 480-506.
6. Asghari, Parvaneh, Amir Masoud Rahmani, and Hamid Haj Seyyed Javadi. "Privacy-aware cloud service composition based on QoS optimization in Internet of Things." Journal of Ambient Intelligence and Humanized Computing (2020): 1-26.
7. Pan, Zixiao, Deming Lei, and Ling Wang. "A knowledge-based two-population optimization algorithm for distributed energy-efficient parallel machines scheduling." IEEE Transactions on Cybernetics (2020).
8. Duc, Thang Le, Rafael García Leiva, Paolo Casari, and Per-Olov Östberg. "Machine learning methods for reliable resource provisioning in edge-cloud computing: A survey." ACM Computing Surveys (CSUR) 52, no. 5 (2019): 1-39.

9. Gabi, Danlami, Abdul Samad Ismail, AnazidaZainal, ZalmiyahZakaria, Ajith Abraham, and NasiruMuhammedDankolo. "Cloud customers service selection scheme based on improved conventional cat swarm optimization." Neural computing & applications (Print) 32 (2020): 14817-14838.
10. Kumar, Mohit, Subhash Chander Sharma, Anubhav Goel, and Santar Pal Singh. "A comprehensive survey for scheduling techniques in cloud computing." Journal of Network and Computer Applications 143 (2019): 1-33.
11. Toosi, Adel Nadjaran, Richard O. Sinnott, and RajkumarBuyya. "Resource provisioning for data-intensive applications with deadline constraints on hybrid clouds using Aneka." Future Generation Computer Systems 79 (2018): 765-775.
12. Zhang, Jie, Zhili Zhou, Shu Li, LeileiGan, Xuyun Zhang, Lianyong Qi, Xiaolong Xu, and Wanchun Dou. "Hybrid computation offloading for smart home automation in mobile cloud computing." Personal and Ubiquitous Computing 22, no. 1 (2018): 121-134.
13. Afzal, Shahbaz, and G. Kavitha. "Load balancing in cloud computing–A hierarchical taxonomical classification." Journal of Cloud Computing 8, no. 1 (2019): 1-24.
14. Juneja, Kapil. "An Intelligent GbMFPA Model for Sales Optimization in Distributed Grid-Market." Wireless Personal Communications 103, no. 3 (2018): 2403-2421.
15. Chakravarthi, Koneti Kalyan, L. Shyamala, and V. Vaidehi. "Cost-effective workflow scheduling approach on cloud under deadline constraint using firefly algorithm." Applied Intelligence 51, no. 3 (2021): 1629-1644.
16. Bansal, Jyoti, and Geeta Rani. "An adaptive fault tolerant scheduling system for desktop grid." EAI Endorsed Transactions on Scalable Information Systems 6, no. 21 (2019).
17. Kari, Tusongjiang, Wensheng Gao, Dongbo Zhao, KaherjiangAbiderexiti, Wenxiong Mo, Yong Wang, and Le Luan. "Hybrid feature selection approach for power transformer fault diagnosis based on support vector machine and genetic algorithm." IET Generation, Transmission & Distribution 12, no. 21 (2018): 5672-5680.
18. Jyoti, Amrita, Manish Shrimali, Shailesh Tiwari, and HarivansPratap Singh. "Cloud computing using load balancing and service broker policy for IT service: a taxonomy and survey." Journal of Ambient Intelligence and Humanized Computing 11, no. 11 (2020): 4785-4814.
19. Milan, Sara Tabaghchi, Lila Rajabion, HamidehRanjbar, and NimaJafariNavimipour. "Nature inspired meta-heuristic algorithms for solving the load-balancing problem in cloud environments." Computers & Operations Research 110 (2019): 159-187.
20. Nadeem, Farrukh. "Using radial basis function neural network to predict dynamic resource availability in heterogeneous distributed environments." Journal of Intelligent & Fuzzy Systems 37, no. 4 (2019): 5619-5632.
21. Kwee, S., P. Raskmark, and S. Velizarov. "Changes in cellular proteins due to environmental non-ionizing radiation. I. Heat-shock proteins." Electro-and Magnetobiology 20, no. 2 (2001): 141-152.
22. Adhikari, Mainak, and Satish Narayana Srirama. "Multi-objective accelerated particle swarm optimization with a container-based scheduling for Internet-of-Things in cloud environment." Journal of Network and Computer Applications 137 (2019): 35-61.
23. Abdulhamid, Shafi'I. Muhammad, Muhammad Shafie Abd Latiff, Syed Hamid Hussain Madni, and Mohammed Abdullahi. "Fault tolerance aware scheduling technique for cloud computing environment using dynamic clustering algorithm." Neural Computing and Applications 29, no. 1 (2018): 279-293.
24. Shishido, Henrique Yoshikazu, Júlio Cezar Estrella, Claudio Fabiano Motta Toledo, and Marcio Silva Arantes. "Genetic-based algorithms applied to a workflow scheduling algorithm with security and deadline constraints in clouds." Computers & Electrical Engineering 69 (2018): 378-394.
25. Tang, Xiaoyong, and Xiaoyi Liao. "Application-aware deadline constraint job scheduling mechanism on large-scale computational grid." PloS one 13, no. 11 (2018): e0207596.
26. Gao, Honghao, Kang Zhang, Jianhua Yang, Fangguo Wu, and Hongsheng Liu. "Applying improved particle swarm optimization for dynamic service composition focusing on quality of service evaluations under hybrid networks." International Journal of Distributed Sensor Networks 14, no. 2 (2018): 1550147718761583.
27. Xiao, Jiuhong, Wenyu Zhang, Shuai Zhang, and Xiaoyu Zhuang. "Game theory–based multi-task scheduling in cloud manufacturing using an extended biogeography-based optimization algorithm." Concurrent Engineering 27, no. 4 (2019): 314-330.
28. Mishra, Kaushik, and Santosh Majhi. "A state-of-art on cloud load balancing algorithms." International Journal of computing and digital systems 9, no. 2 (2020): 201-220.
29. Hayat, Bashir, Kyong Hoon Kim, and Ki-Il Kim. "A study on fuzzy logic-based cloud computing." Cluster Computing 21, no. 1 (2018): 589-603.
30. Dziurzanski, Piotr, Jerry Swan, and Leandro SoaresIndrusiak. "Value-based manufacturing optimisation in serverless clouds for industry 4.0." In Proceedings of the Genetic and Evolutionary Computation Conference, pp. 1222-1229. 2018.
31. Zhang, Hengrun, and Kai Zeng. "Communication-Aware Secret Share Placement in Hierarchical Edge Computing." IEEE Internet of Things Journal (2021).