

# An Heuristic Approach for Efficient Workflow Scheduling in Bioinformatics Cloud

Senathipathi K.<sup>1\*</sup> and Chitra S.<sup>2</sup>

1. CSE, Sri Krishna College of Engineering and Technology, Coimbatore, INDIA
  2. Er. Perumal Manimekalai College of Engineering and Technology, Hosur, INDIA
- \*senathipathik@gmail.com

## Abstract

*In cloud computing, the resources used in computing are shared in place of individual servers or personal devices to handle applications. A major application of cloud is in the domain of bioinformatics where sequencing of genomes require extensive computational power. The architecture in cloud computing comprises delivering the storage, software, infrastructure and services processed by technology in the internet for personnel and institutions that ask for the same. The process by which the job resources are allocated resources in the form of VM or Virtual machines is known as cloud scheduling. Here, a productive as well as a new scheduling algorithm such as the Bacterial Swarm Optimization (BSO) algorithm, min-min algorithm and GA (Genetic Algorithm) have been designed. The effective stochastic exploration methods that imitate the natural system's adaptive process are referred to as the GAs. The application of the GAs is in the areas of several Non deterministic Polynomial (NP)- Hard problems like multi-processor design, task scheduling, optimizing and travelling salesman problem. The algorithm that has been taught the optimal search by means of swarm intelligence that arises because of the competition and the co-operation among group members is the The Bacteria Foraging Optimization Algorithm (BFOA) called BSO algorithm, Global optimization, parallel distributed processing and insensitivity to initial value are some of its benefits. Maximizing resource utilization and minimizing the task expense and completion time is this algorithm's objective. Using e-protein, this algorithm's performance is measured.*

**Keywords:** Cloud Computing, Cloud Scheduling, Min-Min Algorithm, Genetic Algorithm (GA) and Bacteria Foraging Optimization Algorithm (BFOA).

## Introduction

With various gene sequence projects happening around the world, modern biology faces the challenges of data storage and analysis. Efficient computing system can reduce the cost of sequencing significantly. For example about 10 years back human genome sequencing cost about USD one million whereas now it can be achieved for as low as USD 500 due to efficient computing facilities at a low cost. GenBank database of NCBI shows ever increasing cataloguing of

genetic information. Effective computing is also required for matching a genome of a tumor with a matching normal tissue sample which is critical for efficient drug delivery. Cloud infrastructure provides an efficient medium for achieving low cost sequencing and scheduling of tasks plays a crucial role in efficient utilization of resources.

A novel technique to compute is provided using cloud computing. The end users can harness unlimited data and distributed resources of the cloud environment by the cloud providers. The extent to which data can be stored rapidly speeds up in this environment. That is why, balancing the load is a challenge in cloud computing, as the data storage rapidly increases. When many tasks happen simultaneously, the challenge lies in balancing the load. By load balancing, the task is distributed among all the nodes in such a way that a node is neither free nor overloaded. This in turn leads to effective system throughput as the resources are utilized intelligently. This load balancing coupled with appropriate resource allocation tasks is achieved by many load balancing algorithms. The effective allocating of resources in a cloud environment is critical for deriving the maximum benefits [1].

There are 3 service models that are provided by Cloud Computing- Cloud Software as a Service (SaaS), Cloud Infrastructure as a Service (IaaS) and Cloud Platform as a Service (PaaS). Cloud SaaS enables the users to make use of the applications belonging to the provider that run on cloud infrastructures. Accessing the application may be done from different client devices like web browsers, using a thin interface. Cloud IaaS gives the resources in computation such as processing, storage, networks and others. This enables the user to deploy and run the arbitrary software, that includes the Operating Systems and the Applications. Cloud PaaS services facilitates the developers using the Provider particular program language and tools in developing an application. For decreasing costs and maximizing returns, most of the organizations are harnessing cloud computing. So, for influencing the service provider's idea in costs benefits for this computation condition, efficient task scheduling is very important [2].

There are several features of cloud computing- Broad Network Access- Capacities are generally found all over the network I cloud computing. These capabilities can be accessed using different approaches. On Demand Service- As and when the users want the cloud services, they can access the same. Rapid Elasticity- The resource allocation and quantity dynamically changes with demand. Resource pooling- many models for pooling resources are available

that can be used to fulfil the customer demands. Measured service- The service that is provided to a consumer in terms of resources can be monitored and controlled for all the services and also for the providers[3]. The challenges that are involved in cloud computing include requiring fast connectivity, resource scheduling, scale, Quality of Service, security, performance monitoring, consistent and strong service abstractions etc.

Allocating the tasks to the resources that are at disposal based on quality of the tasks, the need and the metrics in such a way that the quality of services by the cloud is not impacted is the foremost challenge in cloud computing which is termed as scheduling. The scheduling scheme involved the technique in which resources such as the I/o, network bandwidth, CPU, secondary storage and primary memory are allocated between the tasks and the users. An efficient scheduler is the one that adapts the strategy for scheduling based on the dynamic environment, the nature of the tasks and the comparative task metrics for a set of tasks that are given for processing. A policy set that regulates the work order to be worked using a computing system is referred to as scheduling. Job scheduling is one of the many types of scheduling algorithms in distributed computing systems. Achieving high computing performance and also the best system throughput is the chief benefit of scheduling. Scheduling provides optimal utilization by managing allocation and utilization of resources such as CPU, memory etc. [4].

The management of dynamic resources on a cloud computing platform is efficiently solved by means of Virtualization technology. By means of agreeing to VM service and by linking it to each physical server, not only the heterogeneous problems as well as platform irrelevance of subscriber's need is met but also SLA or service Level Agreement can be Guaranteed. Furthermore, as per the load change, this technology can remap between the VM and the physical resources for achieving the entire system's load balance dynamically. Hence the use of virtualization is extensive in cloud computing. However, since the resources are dynamically heterogeneous, to achieve its best performance by completely utilizing its resources and services, the VM should dynamically adapt to the cloud computing environment. The resource utility can be improved by load balancing and efficient allocation of resources. Hence an important point of consideration for the research in this area is the way in which the VM resources are scheduled for realizing load balancing and improving the resource utility in cloud computing[5].

There is a requirement for taking into account different Quality of Service or QoS considerations for resource scheduling in cloud computing- System cost, reliability, service completion time and network bandwidth. All these are considered for computing the cloud computing applications. Evolutionary algorithms based on the natural origin of biological species are the ones that can be

employed for scheduling. There are 2 sets of job scheduling algorithms in the distributed computing-.1. Online method of heuristic scheduling algorithm in which the functions are executed individually. The online mode is mainly used in cloud environment [6].2. Batch Mode Heuristic Scheduling algorithm (BMHA) in which the functions are executed as batches e.g. First Come First Serve (FCFS), Min-Min protocol.

Heuristic approaches can be used for solving the NP hard and NP complete problems. Hyper heuristics, meta heuristics and local heuristics are the different heuristic techniques are being made use. For operating in high abstraction level, hyper heuristics are used. For solving hard computational search problems, hyper-heuristics are used that do not need a problem specific knowledge. The hyper heuristics can be performed in the heuristics search space and applied when the nature of resources is dynamic[7]. Meta heuristics are costly methods needing a know-how in both problem domain and heuristic method.

For scheduling in cloud computing, the BSO based work flow scheduling has been proposed and compared with GA and the min min algorithm. The remaining of the research work is scheduled as below: Related works in literature are discussed in the second section. Various methods used in the work are discussed in the third section. Experimental results is at section 4 and work is concluded in section five.

### Related Works

Consideration of various facts which include the bandwidth of the network as well as the auction deadline, Kong et al., [8] presented a new adaptive VM resource scheduling protocol that depend on the method of auction. Firstly, In a given deadline for competition, the client's bids are sequenced. Secondly, the screening of the client group followed by the configuration of the respective VM resources is done such that minimum cost of the Service facilitators are present. At the end, keeping in consideration the range of mean and competing pays, the price of last payment is sorted out, in order to complete the task request of a client with a provided VM source. Outcomes of the simulated experiments show that the QoS of the cloud environment can be increased, along with the increase in the rate of resource utilizing of VM and the cloud provider of service profits, using our put-forth protocol.

Resource provisioning and a scheduling framework that caters to resource scheduling and distribution was proposed by Singh & Chana [9]. Making use of K-means-based clustering protocol, cloud work-loads are being re-clustered. This is done followed by initially clustering with the help of work-load schemes in order that the QoS needs for a work-load is identified, followed by which provisioning the determined need of QoS resources is done prior to the nominal scheduling process. Also, different scheduling policies are made use for scheduling. At the end, the proposed framework was evaluated for performance in real

as well as simulated cloud environment. It was shown by the experimental outcomes that the framework aids the resource scheduling effectively taking into account execution cost, time and energy consumption as QoS parameters.

Min-min algorithm may be unscheduled and this is its downside. The algorithm was improved by Liu et al., [10]. There were three constraints to this algorithm in addition to its improvement- They were- service cost, dynamic priority model and quality of service. For running the simulation experiment, the software CloudSim is used. It has been shown experimentally that it takes a reasonably lesser time for executing longer tasks compared to the conventional min min protocol along with the increasing utilization rate and meeting the requirement of the users.

Un-certainty dependent QoS Min-Min or UQMM protocol was presented by Panda and Jain [11]. This factors the Quality of service based on the uncertain parameters in heterogeneous multi cloud environment. It serves to be the initial technique that compromises the task allotting problem using uncertainty based QoS for a hetero-geneous multicloud systems. By using the benchmark and the synthetic dataset, the author performs extensive simulations and then measures the performance by considering different metrics. The outcomes are then contrasted using smoothing based task scheduling algorithm, min-min scheduling and cloud min-max normalizing for showing how effective the proposed algorithm is.

A new dynamic task scheduling based on improvised GA was proposed by Ma et al. [12]. Based on the GA, the suggested algorithms considers the dynamic features of cloud computation. For simulation, the CloudSim simulation platform is chosen. It has been experimentally proven that executing time of task schedule could be significantly decreased and the cloud system's throughput can be effectively improved using the suggested algorithm.

Cost-Aware Heterogeneous Cloud Memory Model (CAHCM), a new approach was proposed by Gai et al., [13]. This aims to provide a good performing cloud-based heterogeneous Memory Service offerings. The Dynamic Data Allocation Advance (2DA) Algorithm which use genetic programming for determining the data allocations on the cloud-based memories, is the chief algorithm that supports CAHCM. The author, in the suggested approach considers factors that are critical to the performance of cloud memories like data move cost of operation, time constraints, energy performances and communication costs. To remain as a cost-aware cloud dependent solution, the approach has been found to be feasible as well as adaptable.

A new scheduling methodology called PreAntPolicy was proposed by Duan et al., [14]. It contained a prediction model which depend on the Fractal Mathematics along with a scheduler based on improved Ant-Colony algorithm. The scheduler has the onus of scheduling the resources while, under the pretext of ensuring QoS, minimizes the energy

consumption. The prediction model determines if the execution of scheduler has to be triggered by virtue of load trending predicting. whether to trigger the execution of the scheduler by virtue of load trend prediction. It was demonstrated by the performance results, through experimental analysis and simulations that used real workload from the Google's compute clusters, that superior resource utilization and energy efficiency were exhibited by the suggested approach.

The k-means algorithm was improved using the Bacterial Foraging Algorithm (BFA) and Particle Swarm Optimization (PSO), by Sun et al., [15]. The algorithm makes use of enhanced K-means to packet processing tasks, according to the tasks' length and the resource requirement. After this, within the group, it performs Min-Min scheduling algorithm. By means of Cloud Sim platform's simulation and theoretical research, the experimental results peak when there are 300 tasks when compared with the Min-Min, there was a 17.13% improvement in the task completion time.

A new heuristic scheduling protocol to determine more efficient scheduling solutions in cloud computing schemes called the Hyper-Heuristic Scheduling algorithm (HHSA) was developed by Tsai et al., [16]. For dynamically finding out which low level heuristic can be utilized for determining better candidate solutions, the diversity identification and enhancement detecting operators were used by the suggested algorithm. All of the state of art scheduling algorithms are implemented on a simulator which is the CloudSim and a real system, Hadoop. The performance of the suggested method is then compared with these several cutting edge algorithms. It has been shown by the outcome that compared with the other scheduling algorithms, the makespan of task scheduling is significantly decreased by HHSA, on Hadoop as well as CloudSim.

## Methodology

The process of allocating the resource to the incoming jobs for improving their utilization and also enhancing the throughput and cloud performance is referred to as cloud scheduling. There are two levels of cloud service scheduling- system level and user level. The former takes care of the management of the resources within the data centre. The latter solves the issues between service providers and customers. This section presents the description of cloud scheduling using BSO, GA and the min-min algorithm.

**Min-Min Algorithm:** The starting of the scheduling is a set of tasks that are unassigned. Initially, for all the tasks, the least time of completion is determined. From these least times, the one that is the minimum is chosen. This time represents the minimum time for all tasks using any resource. The task is timed to the respective machine based on that minimum time. Then by addition of the time of execution for the task appointed with the time of execution for the additional tasks of that particular machine, updating the time of execution happens. Then, the task that is

allocated is taken off from the task list that should be allocated for the machines. Finally, until all the tasks are allocated on the resources, the same procedure is repeated [17].

**Genetic Algorithm (GA):** The biological tenet of population generation is the basis for GA. It has been regarded as a very quickly growing field in AI. The inspiration for the GA arises from Darwin's theory of evolution. The concept of "Survival of the Fittest", that was suggested by Darwin, is used as a scheduling technique. Here based on the fitness function's value for every attribute of tasking schedule processing, tasks are allocated on the resources. The chief concepts of Genetic Algorithm are explained as given below[18]:

**Initial Population:** The individual set which are utilized in the Genetic Algorithm for finding the optimum solution is the initial population. Individual refers to each solution in the population. For making the individual viable for genetic operations, each of it is represented as a gene. From the original population, operations are applied on selected individuals so that the next generation can be spawned. Based on a certain criteria, the mating genes are chosen.

**Fitness Function:** The fitness value determines the productivity of any individual. It also tells us how superior an individual in a population is. The performance of the individual in a population is represented by the fitness function. Hence, as per the fitness or the function value the individual may continue living else may be considered dead. Thus, in a Genetic Algorithm, the inspiration comes from the fitness function.

**Selection:** Based on Darwin's survival theory, for choosing an intermediate solution for the forthcoming generations, the selection mechanism is used. Based on the performance, the GA is guided by this operation. For selecting the best gene, there are various selection strategies. E.g. the roulette wheel, rank based selection, tournament selection, Boltzmann strategy etc [19].

**Crossover:** This operation is brought forth by choosing two individuals to be the parents. Then, by alternating and reforming the parents' parts, a new individual tree can be created. The search mechanism is boosted by hybridization in the GA which also is the guiding factor.

**Mutation:** Mutation follows crossover. Genetic diversity in the population is introduced by this operator. Whenever, the diversity decreases in a population because of repetitive reproducing and cross-over operating, mutation happens, at the time of evolution based on a low set user-defined mutation likelihood. In the chromosomes, mutation changes one or many values of the genes. A totally new set of gene values can be produced from this and supplemented with the gene-pool. The GA, along with this recent pool of genes may show a better solution compared to before.

### Proposed Bacteria Swarming Optimization (BSO) Algorithm

Prof. K.M. Passino, in 2002, suggested a new computation method that was inspired by nature. This was done by replicating the evolutionary reproduction, environmental elimination dispersal and the food foraging behaviour of *E. coli* bacteria. This came to be known as BFOA, which was also a meta-heuristic protocol as it provided a generic frame-work and some guide-lines to present the problem solution instead of giving detailed analysis and structure. This algorithm adds randomness for simulating natural properties. It incorporates iterations for simulating evolutionary properties. This is being considered by the researchers as a cutting edge paradigm in optimization and it is a relatively new statistical bio-inspired optimization technique. There have been a number of improvisations made. Researchers have also developed many models. It has been successfully used in engineering and other fields. These applications have been documented and for investigating the effectiveness of the optimization algorithm, some theoretical analysis is also being researched [20].

BFOA serves to be another iteration-based optimizing tool, like an evolutionary computational technique. All of the Synthetic or Simulated bacterium are initially located in positions randomly in a multi-dimensional search-space of the problem. Following this, measure the cost or the fitness of the solutions by which global optimum solution is found using these fitness. Following this, particular protocol depending operations are implemented on these separate bacterium so that new solutions are obtained. Repeat these procedures till maximum number of iterations pass or a predefined goal is attained.

The BFOA serves to be non gradient, bio inspired self organizing natural and newly formed effective scheme of optimizing. The replication of the *E. coli* bacteria that survives in the human intestine is done in this algorithm. Of the characteristics that are mimicked are Natural birth-death elimination, food foraging and evolutionary reproduction. The following takes place in a highly complex problem domain- Optimum living fuels that refer to the unit based energy intake is foraged by BFOA. This is a parameter for fitness. Using a foraging behaviour referred to as "Chemotaxis", the bacterium collects this fitness parameter. Fittest bacterium represents the evolutionary based reproduction because of the limited life spans of the bacteria. By reproducing, variation is introduced in the population. The search space is globalized to avoid being trapped in local optima and pre-mature solution. The natural disasters based birth-death depending method of removal and discarding is used. For searching faster, inter communication based social swarming is considered.

Suppose the minimum of  $J(\theta)$  where  $\theta \in \mathbb{R}^p$  (i.e.  $\theta$  is a p-dimensional vector of real numbers) has to be determined, which doesn't have any measures or any analytics

explanation of the gradient  $\nabla J(\theta)$ . The 4 major techniques made use in a real bacteria scheme is mimicked by the BFOA: Chemotaxis, swarming, reproduction, Elimination-dispersal in solving the non gradient problem of optimizing. A Virtual bacteria is considered to be a trial solution (may be called a search-agent) which move over the surface of the function so that the global optima is located [21].

A chemo-tactic step is defined as a tumble which is proceeded by a tumble else tumble chased by run. Let  $j$  be the index in the chemo-tactic step. Also let  $k$  form the index in reproducing step. Have 1 as the index of elimination-dispersing event. Along with them. let

- $p$ : Dimension of the search space,
- $S$ : Total number of bacteria in the population,
- $N_c$ : The number of chemotactic steps,
- $N_s$ : The swimming length.
- $N_{re}$ : The number of reproduction steps,
- $N_{ed}$ : The number of elimination-dispersal events,
- $P_{ed}$ : Elimination-dispersal probability,
- $C(i)$ : The size of the step taken in the random direction specified by the tumble.

Let  $P(j,k,l) = \{\theta^i(j,k,l) | i=1,2,\dots,S\}$  show the location for every member of the entire  $S$  bacterium population in the  $j$ -th chemotacting step,  $k$ -th reproducing step, and  $l$ -th Elimination-dispersing step. Let  $J(i,j,k,l)$  represent the cost in the location of the  $i$ -th bacterium  $\theta^i(j,k,l) \in \mathbb{R}^p$  (few times it drops the index and point to the  $i$ -th bacterium position as  $\theta^i$ ). It is to be noted that it can represent interchangeably  $N$  as “cost” (with the terms from optimization theory) and remaining a nutrient surface (in reference to the biological connections). Considering real population of bacterium,  $S$  is much larger (e.g.,  $S=109$ ), but  $p=3$ . For the computer simulations, it uses small size of population keeping the size of the population to be constant. But BFOA permit  $p > 3$  such that, the technique could be applied to high dimensional problem of optimizing. Followingly, it denotes the 4 main steps in using BFOA.

**Chemotaxis:** The movement of an Ecoli cell is simulated by this process. This movement is simulated using swimming and tumbling using flagellum. There are two different ways in which an Ecoli can move biologically. It can, for sometime, either swim in the same direction or tumble and for its entire period of survival, it can keep alternating between these two movements[22]. Suppose  $\theta^i(j,k,l)$  shows  $i$ -th bacteria at  $j$ th chemotactic,  $k$ -th reproductive and  $l$ -th elimination-dispersal step.  $C(i)$  is the step size considered in random directions mentioned using tumble (run length unit). And during times of computing chemo-taxis, the motion of bacteria is denoted using (1):

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \quad (1)$$

in which  $\Delta$  represents a vector in a direction randomly with elements lying in  $[-1, 1]$ .

**Swarming:** In very active bacteria species such as *S. Typhimurium* and *E. coli*, an interesting group behaviour of the formation of intricate and stable spatio-temporal patterns (swarms) in semisolid nutrient medium has been observed. When located in between a semi-solid matrix using a lone nutrient chemo effected, an E-Coli cell group form in a pattern of traveling ring by shifting the nutrient gradient upwards. During the process of stimulating the cells using a High-level of succinate, an attractive aspartate is released, that aids the bacterium to form a group and to traverse in concentric manner of swarms using large bacteria density. Cell-to-cell signalling of E-Coli swarm is shown as (2).

$$J_{cc}(\theta, P(j,k,l)) = \sum_{i=1}^S J_{cc}(\theta, \theta^i(j,k,l)) \\ = \sum_{i=1}^S [-d_{attractant} \exp(-w_{attractant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] + \\ \sum_{i=1}^S [-h_{repellant} \exp(-w_{repellant} \sum_{m=1}^p (\theta_m - \theta_m^i)^2)] \quad (2)$$

in which  $J_{cc}(\theta, P(j,k,l))$  forms the objective function value to be supplemented to the original objective function (to be minimized) to show a time variant objective function,  $S$  is the entire bacterium quantity,  $p$  is the number of variables to be optimized, that is contained in every bacteria and  $\theta = [\theta_1, \theta_2, \dots, \theta_p]^T$  serves a point in the  $p$ -dimensional searching domain.  $d_{attractant}, w_{attractant}, h_{repellant}, w_{repellant}$  are various co-efficients that is to be chose appropriately.

**Reproduction:** The health state (fitness) for every bacteria is computed followed by every terminated processes of chemo-taxis [23]. The addition of the cost-fuction is (3):

$$J_{health}^i = \sum_{j=1}^{N_c} P^{i,j,k,l} \quad (3)$$

where  $N_c$  serves the entire number of steps for an entire chemo-taxis processes. Location of healthy bacterium show good set of optimizing attributes. For increasing the speed of search and refining the same, many more bacterium should be located in optimization domain areas. The reproductive step takes care of this. 50% of healthy bacteria having the least value of the cost-function is allowed to live and the remaining 50% perish. The population of the bacteria remains unvarying. This is because, each bacteria divides itself into two and both locate in the same place

**Elimination and Dispersal:** A basis for local search is provided by Chemotaxis. The convergence that has been simulated by traditional BFOA is made quicker by the process of reproduction. Chemo-taxis and reproduction alone do not suffice in searching in the global optimum. This

is because the bacteria may get stuck in the starting locations or in the local optima. The diversity presented by the BFOA helps change slowly or to suddenly remove the chances of locked in the local optima. Here, few reproduction process, dispersion takes place. Followed by which few bacterium are selected to be perished depending

on a pre-set probability  $P_{ed}$  or transferred to some other location inside the locality.

The Biotechnology and Biological Sciences Research Council (BBSRC -28/BEP17014 March 2002) aided the e-protein project, through their e-Science programme. The programme objectives are providing annotation of the proteins based on the structure in major genome linking resources at the following locations via cloud technology- University College London (UCL), European Bioinformatics Institute (EBI) and the Imperial College London.

Many model proteomes or "complete" protein set for provided organisms are made public availability. There is huge investment of resources for studying their structural and functional annotation. Based only on the similarity with the other proteins, forty to sixty percent of genome proteins can be forecast. But, higher rates of prediction can be obtained by a more detailed analysis. There are many methods that give in depth analysis- (i)the detection of homologous proteins in which a known structure and function are being determined early [24], (ii) identifying the sequence motifs/profiles which categorise the structure or function.(iii) determining the sequence features like coiled coils and trans membrane areas.

**Results and Discussion**

Experiments were carried out using bioinformatics data obtained synthetically. The proteome is the entire set of proteins expressed by a genome, cell, tissue, or organism at a certain time. Generally, when decided if a protein must be provided with some annotation, a score threshold is set with annotation provided for every proteins thus scoring more compared to threshold value.

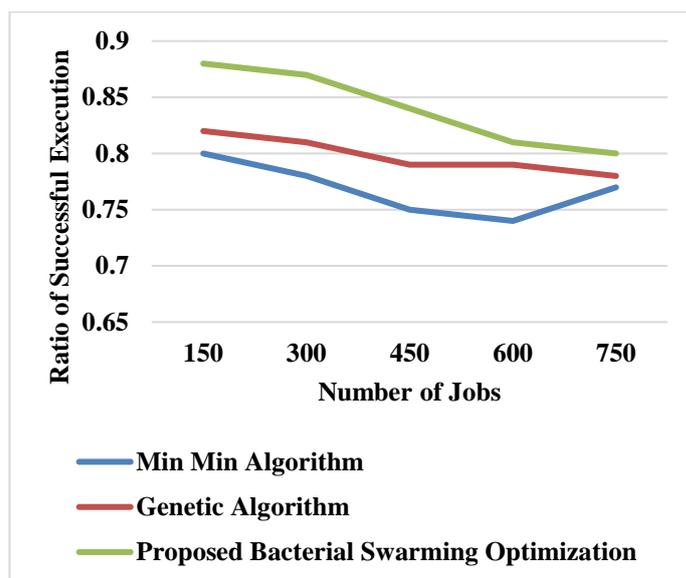
There is a necessity to examine the outcomes of different schemes for producing precise outcomes, as, there may be some errors occurring in annotation. Keeping this in perspective, we should take into account the factor that, obtaining accurate results needs technical problems on the cloud infrastructure and the biological side to be overcome. Proteome annotation is not a well established technique and by comparing the outcomes with various existing annotation schemes serves to be the only way in detecting and resolving issues.

The summary of the e-protein results is shown in Tables 1 and 2. Figure 1 to 5 shows the average schedule length, ratio of successful execution, average schedule length for e-protein, standard deviation for e-protein and ratio of successful execution for e-protein.

**Table 1**  
**Summary of Results**

Number of Jobs	Min-Min Algorithm	Genetic Algorithm	Proposed Bacterial Swarming Optimization
Average Schedule Length in second			
150	321	316	326
300	1020	994	1000
450	1754	1598	1712
600	2334	2290	2427
750	3168	2935	3095
Ratio of successful execution			
150	0.8	0.82	0.88
300	0.78	0.81	0.87
450	0.75	0.79	0.84
600	0.74	0.79	0.81
750	0.77	0.78	0.8

It is noted that the proposed BSO shows higher average schedule length by 1.54% & 3.11% for 150 number of jobs, by 1.98% & 0.6% for 300 number of jobs, by 2.42% & 6.88% for 450 number of jobs, by 3.9% & 5.8% for 600 number of jobs and by 2.33% & 5.3% for 750 number of jobs when compared with min-min algorithm and GA.

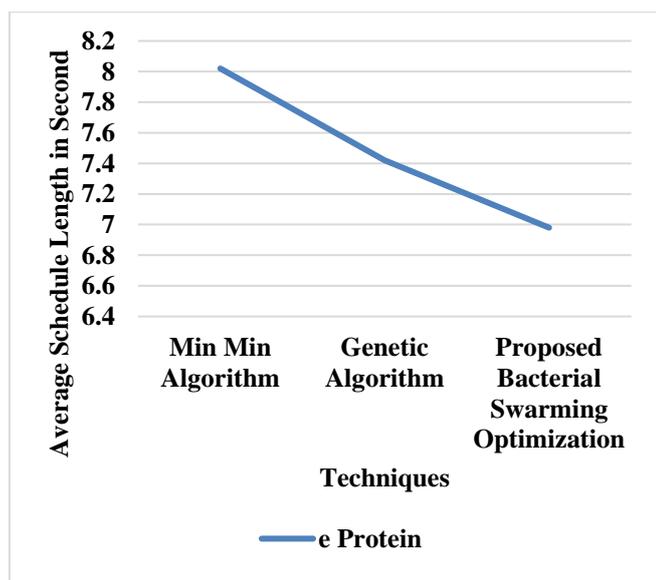


**Figure 1: Ratio of Successful Execution**

Using figure 2, it is noted that the proposed BSO shows high ratio of successful execution about 9.52% & 7.05% for 150 number of jobs, by 10.9% & 7.14% for 300 number of jobs, by 11.32% & 6.13% for 450 number of jobs, by 9.03% & 2.5% for 600 number of jobs and by 3.82% & 2.53% for 750 number of jobs when compared with min-min algorithm and GA.

**Table 2**  
**Summary of Results for e Protein**

Techniques	Min-Min Algorithm	Genetic Algorithm	Proposed Bacterial Swarming Optimization
Average Schedule Length in second			
e Protein	8.02	7.42	6.98
Standard Deviation			
e Protein	0	0.12	0.08
Ratio of successful execution			
e Protein	78	84	92



**Figure 2: Average Schedule Length in Second for e Protein**

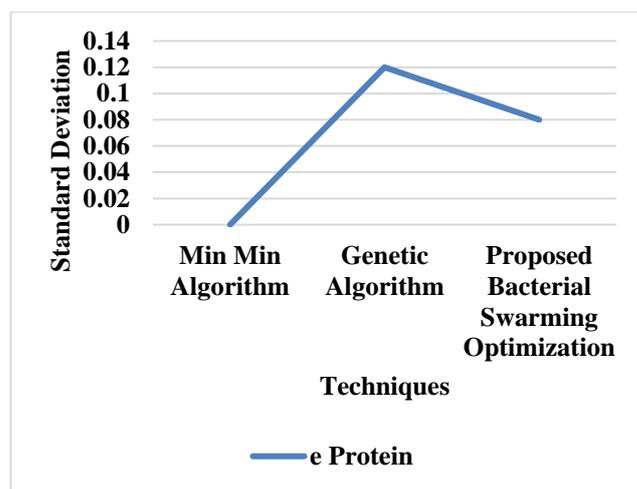
Using figure 3, it is noted that, the proposed BSO shows lesser average schedule length by 13.86% & 6.11% for e protein when compared with min-min algorithm and GA.

Using figure 4, it is noted that the proposed BSO shows lesser standard deviation by 200% & 40% for e protein when compared with min-min algorithm and GA.

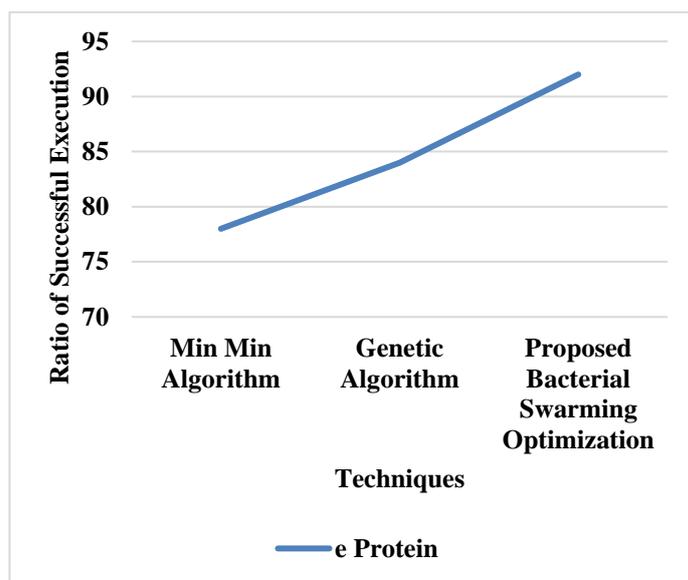
Using figure 5, it is noted that the proposed BSO shows greater ratio of successful execution by 16.47% & 9.09% for e protein when compared with min-min algorithm and GA.

**Conclusion**

The concept of cloud computing is based on the internet wherein depends on the demand, the resource allocation to users is performed. The most important part of the cloud computing environment is the task of scheduling. The author, in the current work, has developed and formulated optimal cloud scheduling policies like BFOA, GA and min-min.



**Figure 3: Standard Deviation for e Protein**



**Figure 4: Ratio of Successful Execution for e Protein**

In places where the search space is large and irregular with the requirement of the global optimum, GA has been found to be the most useful. There are 4 processes that govern the foraging behaviour of the Ecoli- Chemotaxis, reproduction, Elimination and Swarming. It has been shown by the results that proposed BSO has higher ratio of successful execution by 3.82% & 2.53% for 750 jobs, by 9.03% & 2.5% for 600 jobs, by 11.32% & 6.13% for 450, by 10.9% & 7.14% for 300 jobs and by 9.52% & 7.05% for 150 jobs visa vis the GA and the min-min algorithm. Compared to these two, the proposed BSO has a successful execution rate higher by 16.47% & 9.09% for e protein.

**References**

1. Kaur, R., & Luthra, P. (2014). Load Balancing in Cloud System using Max Min and Min Min Algorithm. *algorithms*, 10(11), 12.
2. Kaur, S., & Verma, A. (2012). An efficient approach to genetic algorithm for task scheduling in cloud computing environment. *International Journal of Information Technology and Computer Science (IJITCS)*, 4(10), 74.

3. Kaur, R., & Luthra, P. (2012, December). Load balancing in cloud computing. In *Second Symposium on Cloud computing. International Conference on Recent Trends in Information, Telecommunication and Computing*, 374-381.
4. Varma, P. S., & Krishna, S. R. (2016). Design and Development of Efficient Cloud Scheduling Algorithm based on Load Balancing Analytics. *International Journal of Computer Applications*, 138(12).
5. Hu, J., Gu, J., Sun, G., & Zhao, T. (2010, December). A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. In *Parallel Architectures, Algorithms and Programming (PAAP), 2010 Third International Symposium on* (pp. 89-96). IEEE.
6. Gupta, R. O., & Champaneria, T. (2013). A Survey of Proposed Job Scheduling Algorithms in Cloud Computing Environment. *International Journal of Advanced Research in Computer Science and Software Engineering*.
7. Jacob, L., Jeyakrishnan, V., & Sengottuvelan, P. (2014). Resource scheduling in cloud using bacterial foraging optimization algorithm. *International Journal of Computer Applications*, 92(1).
8. Kong, W., Lei, Y., & Ma, J. (2016). Virtual machine resource scheduling algorithm for cloud computing based on auction mechanism. *Optik-International Journal for Light and Electron Optics*, 127(12), 5099-5104.
9. Singh, S., & Chana, I. (2016). Resource provisioning and scheduling in clouds: QoS perspective. *The Journal of Supercomputing*, 72(3), 926-960.
10. Liu, G., Li, J., & Xu, J. (2013). An improved min-min algorithm in cloud computing. In *Proceedings of the 2012 International Conference of Modern Computer Science and Applications* (pp. 47-52). Springer Berlin Heidelberg.
11. Panda, S. K., & Jana, P. K. (2016). Uncertainty-Based QoS Min-Min Algorithm for Heterogeneous Multi-cloud Environment. *Arabian Journal for Science and Engineering*, 41(8), 3003-3025.
12. Ma, J., Li, W., Fu, T., Yan, L., & Hu, G. (2016). A novel dynamic task scheduling algorithm based on improved genetic algorithm in cloud computing. In *Wireless Communications, Networking and Applications* (pp. 829-835). Springer India.
13. Gai, K., Qiu, M., & Zhao, H. (2017). Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing. *IEEE transactions on cloud computing*.
14. Duan, H., Chen, C., Min, G., & Wu, Y. (2017). Energy-aware scheduling of virtual machines in heterogeneous cloud computing systems. *Future Generation Computer Systems*, 74, 142-150.
15. Sun, X. Y., Fu, X. L., Hu, H., & Gui, T. (2014). The Cloud Computing Tasks Scheduling Algorithm Based on Improved K-Means. In *Applied Mechanics and Materials* (Vol. 513, pp. 1830-1834). Trans Tech Publications.
16. Tsai, C. W., Huang, W. C., Chiang, M. H., Chiang, M. C., & Yang, C. S. (2014). A hyper-heuristic scheduling algorithm for cloud. *IEEE Transactions on Cloud Computing*, 2(2), 236-250.
17. Kumar, P., & Verma, A. (2012, August). Scheduling using improved genetic algorithm in cloud computing for independent tasks. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics* (pp. 137-142). ACM.
18. Hamad, S. A., & Omara, F. A. (2016). Genetic-Based Task Scheduling Algorithm in Cloud Computing Environment. *International Journal of Advanced Computer Science & Applications*, 1(7), 550-556.
19. Goyal, T., & Agrawal, A. (2013). Host scheduling algorithm using genetic algorithm in cloud computing environment. *International Journal of Research in Engineering & Technology (IJRET)* Vol, 1, 7-12.
20. Mahapatra, G., & Banerjee, S. (2013). A study of bacterial foraging optimization algorithm and its applications to solve simultaneous equations. *International Journal of Computer Applications*, 72(5).
21. Das, S., Biswas, A., Dasgupta, S., & Abraham, A. (2009). Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications. In *Foundations of Computational Intelligence Volume 3* (pp. 23-55). Springer Berlin Heidelberg.
22. Chen, C. H., Su, M. T., Lin, C. J., & Lin, C. T. (2014). A Hybrid of Bacterial Foraging Optimization and Particle Swarm Optimization for Evolutionary Neural Fuzzy Classifier. *International Journal of Fuzzy Systems*, 16(3), 422.
23. Li, J., Dang, J., Bu, F., & Wang, J. (2014). Analysis and improvement of the bacterial foraging optimization algorithm. *Journal of Computing Science and Engineering*, 8(1), 1-10.
24. O'Brien, A., Newhouse, S., & Darlington, J. (2004, September). Mapping of scientific workflow within the e-protein project to distributed resources. In *UK e-Science All Hands Meeting* (pp. 404-409).