Webology (ISSN: 1735-188X)
Volume 17, Number 2, 2020

An Approach For Task Scheduling In Cloud Computing Using Hybrid Algorithm

Sachin V. Solanki

Asstt. Professor, Department of Information Technology, K.D.K.
College of Engineering, Nagpur, India.

Abstract

The persistent exponential development of the Internet and World Wide Web, always improving high-bandwidth communications, broad spread accessibility of influential computers and low-cost components will promote improve transformation of computational grids to an actuality. Today, the latest paradigm to come into view is that of Cloud computing which assures steady services delivered through next-generation data centers that are built on compute and storage virtualization technologies. The Cloud seems to be a particular point of access for all the computing requirements of users. Virtualization is an expertise that hides away the facts of physical hardware and offers virtualized resources for high-level applications. A virtualized server is usually called a virtual machine (VM). The service, known as "Cloud service," is passed on through data centers that are built on virtualized computation and storage technologies.

To pick the most capable service from the fittest service sets, task Scheduling based on Hybrid Algorithm is presented in this paper as research work for the Cloud Computing Environment using CloudSim Simulator. Proposed approach is an optimized approach by hybridizing genetic algorithm and tabu search algorithm.

In Cloud environment, VM provisioning is carried out at two levels: first, at the host level and second, at the VM level. At host level, scheduling is to schedule and map the virtual machine (VM) resources to physical machines (or hosts) in order to find the best mapping solution to meet the system load balance to the greatest coverage. At the VM level, the VM consigns a fixed amount of the available processing power to the individual application services (task units) that are deployed within its execution engine.

Keywords: Genetic Algorithm, Tabu Search, Physical Machine.

1. Introduction

Cloud computing is certainly the most prospective new standard, due to its potentiality of creating everything more striking as a service. Cloud computing services should be enormously accessible, scalable, and autonomic to carry ubiquitous access, dynamic discovery and composition. In such an intricate environment, how to compose QoS provisioning for various user requests is a challenge. In Cloud Computing, QoS is the standard which estimates the accomplishment of users using the Cloud Computing services. Similarly, Cloud services with the same functions will be accessible by different data centers/Cloud providers. Hence, the preference of the service from the existing service sets attached to unusual data centers and to devise resource provisioning should be considered to assure QoS requests. However, in Cloud computing many imperative problems need to be worked out for the realization of the fine scenery which theoretically illustrates cloud computing. Task scheduling system is one of the center and challenging issues in a Cloud Computing system. In fact, a task scheduling system performs a very important role in how to achieve Cloud computing users' job QoS requirements and use the Cloud resources competently in an economic way. To select the most competent service from the available service sets, task Scheduling based on Hybrid Algorithm (hybridization of Genetic Algorithm with Tabu Search) is proposed as research work for the Cloud Computing Environment using CloudSim.

The CloudSim [69] toolkit backs both system and performance modeling of Cloud system components such as virtual machines (VMs), data centers and resource provisioning policies. It employs generic application provisioning methods that can be further extended with easiness and partial efforts.

Academicians, researchers and industry-based engineers can analysis the performance of a lately developed application service in a managed and easy to set-up environment.

Generally, it is the intention of the scheduling system to assign resources to a definite application request for a certain amount of time. In the context of this work scheduling deals with the allocation of resources to user requests for mostly computational applications. These requests are usually called tasks (in CloudSim, Cloudlets).

A task scheduling is a process that manages and maps the execution of tasks on the hosts in data centres. It allocates appropriate tasks to hosts so the execution is often completed to assure objective functions imposed by users. The common concern in scheduling tasks on distributed resources belongs to a class of issues called NP-hard issues. For this sort of problems, it is complex to find algorithms to produce the optimal solution within polynomial time. Even if the task scheduling problem can be solved by using extensive search, the methods complexity for solving task scheduling is very high. In Cloud environments, scheduling decisions must be taken in the least time possible, because there are many users computing for resources, but at any time, time slots required by one user could be taken by another user. Paper discusses various approaches for task scheduling problem. Hybrid approach combines Genetic Algorithm (GA) and Tabu Search (TS) which runs the GA as the major algorithm and calls TS method to enhance individuals of the population.

In solving hard problems where there requires proficient exact solutions, heuristic methods, including genetic algorithms (GAs) and tabu search, have grown to be admired substitutes. GAs can be straightforwardly parallelized to balance its computing capability because of its intrinsic parallelism, and hence propose immense potential toward solving hard problems. GAs represent competent solutions by strings of symbols, or linear chromosome, and simulate the development of natural selection, crossover, and mutation among a population of chromosomes. Fitness parameters of chromosomes are assessed based on the excellence of the solutions they signify, and the fitter chromosomes are specified superior probability of survival and reproduction. All scheduling problems are NP-hard problems in which the complexity and time required to resolve the problem raise with the problem size.

2. Review of Literature

Task scheduling system is responsible to choose the finest appropriate jobs in a Cloud for Cloud computing users' requests, by considering some parameters constraints. The mass of research study in Grid Computing can be applied directly in Clouding Computing environment. Today, many research works can be found that have done on Job scheduling in Grid computing and have discussed a larger outlook for the roles of job scheduling in a Grid computing situation [4-10]. The presented topologies of job scheduling system in Cloud or Grid are classified into centralized and decentralized schedulers [4]. Zhiguang, S. and L. Chuang [5] focused a concise clarification of a modeling and pursuance assessment of hierarchical job scheduling. Bayati [6] discussed an iterative scheduling technique on the grids. Patel Y. and J. Darlington [7] introduced a novel stochastic method for QoS-constrained workflows job scheduling in a web service-oriented grid.

Recently, many academic researchers began to work on the QoS of job scheduling system [9-13]; that set ahead the way of QoS performance analysis for Cloud Computing services with dynamic scheduling system. Two feedback dynamic scheduling technique for scheduling mechanism has been discussed in [18]. Masnida Hussin [19] showed how the diversity of jobs features such as unstructured/unorganized arrival of jobs and priorities, could direct to incompetent job allocation. Prasad, Faruquie, Subramaniam and Mohania [29] presented an approach for job allocation for data processing services over the cloud considering amongst others the processing power, and memory needs. The modeled system [31] implemented a particular case of parallel job scheduling called Gang Scheduling in which jobs having tasks that should be scheduled to execute concurrently and parallel as they are in recurrent communication with each other. This requires a one-mapping between tasks and VMs, and avoids possible bottlenecks or deadlocks, because of tasks waiting for input from other tasks that are not performing. Optimization procedures requirements featuring on numerous parameters are emphasized in [32-33]. Many approaches have been used to explain re-entrant flow shop scheduling problem where in the job of Danping and Lee [35] has a detailed explanation of this problem and the related methodology are proposed. Jiahui Jin [38] proposed a heuristic task scheduling algorithm called BAlance-Reduce (BAR), in which an early task allocation will be formed at first, then the job completion time can be reduced gradually by tuning the early task allocation.

GA is capable of narrowing the search area around the required decision in a short time. However, because of stochastic character of search approach, completing the task can acquire considerable amount of time. Moreover, in scheduling tasks the early information is represented as sets of discrete elements, which are connected with each other in non-trivial way. A number of studies have been devoted to methods of increasing of GA effectiveness. D. Bogdanovich, A. Ziyangirov [47] represents grouping of GA and "traditional" search techniques (e.g. methods of gradient search multidimensional minimization and so on). GAN Guo-ning, HUANG Ting-Iei, GAO Shuai [48] introduces an optimized method for task scheduling foundation on genetic simulated annealing algorithm in cloud computing and its accomplishment. Technique considers the QOS needs of different type tasks, the QOS parameters are handled with dimensionless. The technique proficiently finishes tasks scheduling in the cloud computing environment computing. HUANG Qi-yi, HUANG Ting-lei proposed [49] a job scheduling strategy and approach based on QoS, which could gather user requirements on time and cost. But in the scheduling, the communication between the tasks and the cost of the tasks waiting in the queue are not considered.

Lizheng Guo, Shuguang Zhao1, Shigen Shen1, Changyuan Jiang [57] discussed an optimized algorithm based on the Fuzzy-GA optimization which considers a scheduling assessment by executing the absolute group of task in the job queue. V. Krishna Reddy and Dr. L.S.S. Reddy [58] presented a comprehensive study of methods, frameworks and algorithms used for management and scheduling of data-intensive application workflow in cloud computing environment.

Computational Grid (Grid Computing) is a novel representation that coerces the computing arena in the new millennium. Unification of universally remote and assorted jobs united with the rising computational needs for Grand Challenge Applications (GCA) and enhanced expansion of the Internet and communication technology added fuel the extension of global computational power grids. Ajith Abraham, Rajkumar Buyya and Baikunth Nath [62] attempted to address the dynamic scheduling of jobs to the geographically scattered computing jobs. A short description of the three nature's heuristics namely Genetic Algorithm (GA), Simulated Annealing (SA) and Tabu Search (TS) has been discussed and further demonstrated the hybridized usage of the above methods that can be applied in a computational grid environment for job scheduling. Salim Bitam [63] presented a novel Bee Swarm optimization algorithm called Bees Life Algorithm (BLA) used to proficiently schedule computation jobs among executing jobs onto the cloud datacenters. It is a NP-Complete problem and it aspires at dispersion the workloads among the processing jobs in an optimal manner to reduce the whole execution time of jobs and then, to grows the effectiveness of the whole cloud computing services. Rajkumar Buyya [69] presented cloud simulator "CloudSim toolkit" which maintains both system and behavior modeling of Cloud system components such as virtual machines, data centers and resource provisioning policies.

3. Design of a Scheduling System

Task scheduling system is accountable to pick the best suitable resources in a Cloud for Cloud computing users' jobs, by taking some static and dynamic parameters restrictions of Cloud computing users' jobs into consideration.

From a systemic viewpoint of a Cloud Computing environment, it is considered to acquire a Cloud Computing environment as an enormously powerful server. This server will serve the Cloud computing users' jobs. For each Cloud computing user may has unrelated QoS requirement, usually, Cloud computing users' jobs have different priorities to be practiced.

The main purpose of scheduling is to schedule tasks to the adaptable virtual machines and that to hosts in accordance with adaptable time, which in fact involves finding out a proper sequence in which tasks can be executed under transaction logic constraint.

The scheduling process is responsible for the above mentioned allocation of resources to a user request. This includes the task of finding suitable resources for such a request, deciding which resources to chose from the actual available and when to start the particular application. As there are usually many requests submitted while there are only a limited amount of resources, this causes resource conflicts which must be managed by the scheduling algorithm. This is done under the principle to cause the most proficient schedules. The design and assessment of such a scheduling system is subject of this work.

Task scheduling is considered as one of the most famous combinatorial optimization problems. Scheduling can be done at two levels. At first level, mapping relationship between virtual machines (VMs) and physical machines (PMs) can be described. The set of all the physical machines in the system is PM = {PM1, PM2 ,…, PMn}, n is the number of physical machines, PMi ($1 \leq i \leq n$) refers to physical machine number i. Virtual machines (VMs) set on physical machine PMi VMi = {VMi1, VMi2,..,VMim) in which im is the number of VMs on physical machine number i. Suppose in order to allocate virtual machine VM at present, and it uses S = (S1, S2, .., Sn) to indicate the mapping solution set after VM is set to every physical machine. Si here refers to the mapping solution when virtual machine VM is set to physical machine PMi.

At second level, consider Jn tasks n={1,2,….N} on Rm virtual machines m={1,2,….,M} (which are already scheduled at first level) with an objective of minimizing the completion time and utilizing the resources efficiently. Any task Jn has to be processed in Rm, until completion.

4. View on Heuristic Algorithms for Scheduling Problem

4.1 Genetic Algorithm

Genetic algorithms are probabilistic Meta heuristic process, which may be used to enlighten optimization problems. Gas are capable to "evolve" solutions to realistic problems, if they have been correctly encoded. A probable solution to a problem may be characterized as a set of parameters. These parameters (identified as genes) are coupled together to form a string of values (identified as a chromosome). The particular values the genes signify are called its alleles. The place of the gene in the chromosome is its locus. A fitness function must be devised for each problem to be solved. For a particular chromosome, the fitness function returns a single numerical fitness, which will decide the capability of the individual, which that chromosome corresponds to. It begins with the preliminary solution called Population and it is filled with chromosome.

The formation of a genetic algorithm for the scheduling problem can be categorized into four sections: The option of representation of individual in the population; the evaluation of the fitness function; the construction of genetic operators; the formation of probabilities controlling the genetic operators.

Pseudocode for GA

Step 1 Generate initial population.

Encoding

Problem (chromosome) representation is particularly very important and it directly controls the performance of the proposed algorithm while applying genetic algorithm. The first preference a designer has to plan is how to imply a solution in a chromosome.

In GA technique, every solution is encoded as a chromosome. Each chromosome has N genes, as chromosome length.

A population is a set of chromosomes (or individuals) and each signifies a possible solution, which is a mapping sequence between virtual machines and physical machines. The preliminary population can be produced by other heuristic algorithms. To understand Genetic Algorithm, a simple generalized example is elaborated. Here, each chromosome has number of genes and its corresponding fitness value. One chromosome (or individual) can be represented initially as

1	2	3	4	5	6	7	8	9	10	11	12
5	1	3	5	2	1	3	5	4	2	3	4

Figure 1 Initial Solution

Here it is assumed that if there are five physical machines (PM) and twelve virtual machines (VM) and virtual machines are to be allocated to physical machines. The initial solution is shown in the figure 1 representing VM1 allocated to PM5, VM2 allocated to PM1, VM3 allocated to PM3, VM4 allocated to PM5 and so on.

Step 2 Estimate population

Each chromosome is coupled with a fitness value. The objective of GA search is to locate the chromosome with optimal fitness value. For this implementation fitness of individual candidate is calculated by measure of utilized values of physical and virtual machines (generalized, depending on the problem).

Ex. Suppose there are two PMs P1 and P2 having capacity as 50 and 100 units respectively.

There are three VMs V1, V2 and V3 having capacity 10, 20 and 30 units. Supposed all VMs are allocated to P2 then only one PM (i.e. P2) is utilized and utilized value will be as

UtilizedValue = ((10/100) + (20/100) + (30/100))/3

Here UniqueCount is 1 as one PM used.

So, Fitness = UtilizedValue/UniqueCount

Step 3 Perform Crossover to produce offspring.

Crossover operation selects a random pair of chromosomes and chooses a random point in the first chromosome. Roulette Wheel selection operator is considered here for selecting a pair of chromosomes. A crossover operator is used to recombine two strings to obtain a better string. In crossover operation, recombination procedure makes different individuals in the consecutive generations by combining material from two individuals of the previous generation. In selection of reproduction, superior strings in a population are probabilistically allocated a larger number of copies. It is vital to note that no new strings are formed in the reproduction phase. In the crossover operator, new strings are formed by exchanging information among strings of the mating pool. The two strings taking part in the crossover operation are identified as parent strings and the resulting strings are recognized as children strings [71].

For this example crossover operator is applied at the mid of the string.

Ex. Suppose two chromosomes, selected using roulette wheel selection operator, are shown as

C1: {5,1,3,5,2,1,**3,5,4,2,3,4**}

C2: {**4,3,2,5,1,3**,5,2,4,2,4,1}

After performing the crossover, two children are generated as
C1: {5,1,3,5,2,1,5,2,4,2,4,1}
C2: {**4,3,2,5,1,3,3,5,4,2,3,4**}

Step 4 Perform Mutation to offspring.

Mutation adds new information in an arbitrary way to the genetic search procedure and ultimately supports to avoid getting trapped at local optima. It is an operator that begins diversity in the population whenever the population tends to become uniform due to repeated utilization of reproduction and crossover operators. Mutation in a form is the procedure of arbitrarily disturbing genetic information.

Step 5 Choose parents and offspring to form the new population for the next generation.
Lastly, the chromosomes from this modified population are estimated again. This finishes one iteration of the GA. The GA ends when a predefined number of estimations are reached.

Step 6 If terminate condition is achieved finish, otherwise go to Step 2.

4.2 Tabu Search

Tabu Search (TS) is a meta-heuristic approach used to work out combinatorial optimization problems. The fundamental principle of TS is to follow Local Search whenever it encounters a local optimum by permitting non-improving moves; proceeding back to earlier visited solutions is prohibited by the use of memories, called tabu lists that store the current history of the search. TS algorithm beginning from predefined solution and iteratively produce a new solution through its neighborhood. In TS acceptance of proceeding to next solution in neighborhood is deterministic. To avoid cycling and promote larger movement through the answer space, a tabu list is kept of incomplete or absolute solutions. It's not allowed to move to a solution that contains components of the tabu list. It is one of the most efficient local search algorithms for task scheduling problems. It consists of the tabu list, aspiration criteria, neighborhood structures, the move attributes and stopping rules.
Tabu List (TL) is maintained by the tryout solutions in the order in which they are made. Each time a fresh element is added to the end of a list, the oldest element on the list is taken out from the 'top'. Empirically, TL sizes which offer superior results often grow with the size of the problem and stronger restrictions are usually joined with smaller size. The size of the tabu list is primarily assigned according to the size of the problem and it will be reduced and enhanced during the formation of the solution so as to get better exploration of the search space. Tabu Search algorithm has been chosen to utilize in Cloud Computing using multiple tasks on restricted resource. The most significant function in TS is neighbourhood function. The neighbourhood function is a mapping which describes for each solution y a subset M(y) of solutions called neighbourhood. Each solution of M(y) is called a neighbour of y. A local search algorithm begins from some predefined solution and proceeds from neighbour to neighbour as long as possible while reducing the objective of function value.
A general aspiration criterion is better fitness value, i.e. the tabu category of a move in the tabu list is overridden if the move generates an enhanced solution.

General steps for Tabu Search
Step 1 Produce predefined solution y.
Step 2 Form the Tabu List.
Step 3 While set of candidate solutions Y" is not complete.
Step 3.1 Produce candidate solution y" from existing solution y
Step 3.2 Insert y" to Y" only if y" is not present in tabu or if one of the Aspiration Criterion (ex. fitness value) is achieved.
Step 4 Choose the finest candidate solution y* in Y".
Step 5 If fitness(y*) > fitness(y) then y = y*.
Step 6 Refresh Tabu List and Aspiration Criteria
Step 7 If finishing condition met, otherwise go to Step 3.

5. Hybrid Algorithm

Need for the Hybrid Algorithm

Conventional understanding in the area of scheduling is that scheduling problems demonstrate such prosperity and diversity that no particular scheduling technique is adequate. Heuristics resultant from the nature has established a surprising degree of efficiency and simplification for dealing combinatorial optimization problems. Meta-heuristics is used to explain with the computationally tough optimization problems. Meta-heuristics made up of a high level technique that directs the search using other meticulous approaches. Meta-heuristics are used as an individual method for solving tough combinatorial optimization problems. But now the single technique is significantly altered and consideration of researchers has moved to think different type of high level algorithms, namely hybrid algorithms. There are two main concerns have to be considered while coupling the more than one meta-heuristics:
(a) how to decide the meta-heuristic approaches to couple and (b) how to couple the decided heuristic approaches into new hybrid method. For the first concern, various classes of search algorithms can be thought for the intentions of hybridization, like exact methods, simple heuristic methods and meta-heuristics. Likewise, meta-heuristics themselves are divided into local search based methods, population based methods and other classes of nature motivated meta-heuristics. Hence, in general, anyone could couple any technique from the same class or techniques from different classes.

Local search methods (ex. Tabu search) and genetic algorithms are frequently viewed as two complementary tools. A local search algorithm's capability to trace local optima with high precision complements the skill of genetic algorithms to capture a global view of the search space. The genetic algorithm can be considered as a pre-processor for evaluating the initial search, before moving a local search approach to optimize the final population. Tabu search should be used for continuous search spaces as local search process to enhance the quality of the solutions generated by a genetic algorithm in order to answer various a real-world problems. Doing local search on a genetic algorithm's population can initiate diversity and assist to oppose the genetic drift. It facilitates reasonable representation of various search areas in order to handle untimely convergence.

The effectiveness of a local search in achieving a local optimum integrates the competence of a genetic algorithm in isolating the most capable basins of the search space. Hence, integrating a local search into a genetic algorithm can produce in a proficient algorithm. The competence of the search can be enhanced in terms of the time needed to accomplish the global solution, and/or the memory needed to route the population.

GA is a population based method. It needs evaluation of populations over generations. For difficult problems this type of process results in a great computational effort. TS, allowing neighboring moves and not needing objective function gradient data as GA, are well-organized. Though TS uses deterministic moves which decreases unpredictability due to initial solutions and other parameters.

Development of TS is an iterative process, thus hoping to find the best solution at the very beginning would be naive. GA appears to perform well in an environment when information is limited. It seems like the longer the solution time the better the probability that TS will show superior performance. Local-search component and constraint handling flexibility of TS makes it attractive for problems having many constraints.GA is good at performing global search and TS is effective for fine tuning for the scheduling problem.

The objective of the proposed hybrid algorithm to minimize

- Execution Time
- Resource Utilization
- System Cost
- Power Consumption
- Completion Time

Genetic algorithm is capable of doing a parallel search to discover the global search space. Through the parallel search mechanism GA retains useful information about what has been learned from previous generations. GA searches the solution from a population of points instead of a single point. The algorithm is computationally simple and powerful. Tabu Search (TS) works on the individual string, which are points on the solution space. TS guides the iterations from one neighborhood point to another by locally improving the solution's quality and has the ability to avoid poor local minima. Integration of GA and TS using their own strengths has a good chance of providing a reasonable solution to global combinatorial optimization problems such as Task Scheduling. During the hybrid search process, GA starts with a set of initial solution and generates a set of new solutions. On each set of new solution, TS performs a local search to improve them. Then GA uses the improved solution of TS to continue with parallel evolution.

Hybrid Algorithm (HA) steps

To formulate the algorithm, the PM lists and VM lists are been considered.

1 Encoding / Generate initial population

This approach is to dynamically generate an optimal schedule so as to complete the tasks in a minimum period of time as well as utilizing the resources in an efficient way.

For applying GAs directly or coupled with other meta-heuristics, problem (chromosome) representation is very significant and it directly affects the performance of the proposed algorithm. The first decision a designer has to take is how to represent a solution in a chromosome.

To understand the hybrid algorithm, a simple generalized example is taken to schedule seven virtual machines to four physical machines.

PM0 has capacity 100 units
PM1 has capacity 300 units
PM2 has capacity 900 units
PM3 has capacity 100 units
And
VM0 has capacity 50 units
VM1 has capacity 400 units
VM2 has capacity 200 units
VM3 has capacity 100 units
VM4 has capacity 50 units
VM5 has capacity 300 units
VM6 has capacity 100 units

In Cloud environment, each host (or Physical machine) models a physical resource such as a compute or storage server. It contents significant information such as the amount of memory and storage, a list and type of processing cores (to represent a multi-core machine), an allocation of policy for sharing the processing power among virtual machines, and policies for provisioning memory and bandwidth to the virtual machines.

VM represents a virtual machine, which is handled and hosted by a Cloud host component. Every VM component has access to a component that stores the following characteristics related to a VM: accessible memory, processor, storage size.

In GA method, every solution is encoded as a chromosome. Each chromosome has N genes, as chromosome length. A schedule has appeared in form of a solution or in other word, a chromosome. The below figure, depicts a chromosome.

3	2	1	0	2	3	1
V0	V1	V2	V3	V4	V5	V6

Figure 2: Chromosome representation

Here, VM0 is mapped to PM3, VM1 is mapped to PM2, VM2 is mapped to PM1 and so on.

2 Declaration of Tabu List

Declaration of Tabu list for PMCount = 4 and VMCount = 7
TabuList = [VMCount][PMCount]

3 Chromosome evaluations

Each chromosome is associated with a fitness value. The goal of GA search is to find the chromosome with optimal fitness value.

Fitness value of each individual is calculated by considering terms UtilizedValue and UniqueCount.

UtilizedValue = VMCapacityi / PMCapacityj
For i = 0,1,…n VMs and j = 0,1,2,..m PMs

UniqueCount is the number of PM used and so, in this case all four PMs are utilized
So, UniqueCount = 4
Fitness = UtilizedValue / UniqueCount

Better the fitness value better the approach.

4 Update the TabuList based on the fitness value

5 Fitness of all individuals is calculated and initialized (including best individual having best fitness value)

6 for number of iterations

a. Apply RouletteWheelSelection to arrange all individual in the population
b. Select two individuals from the RouletteWheelList
c. Apply Crossover
   In Crossover, only uncommon genes are taken
   for further processing.

3	2	1	1	2	2	2
V0	V1	V2	V3	V4	V5	V6

Figure 3: First Individual

2	1	2	1	3	2	2
V0	V1	V2	V3	V4	V5	V6

Figure 4: Second Individual

Uncommon Genes are 0 1 2 4

d. Apply mutation on offsprings using uncommon
   genes.

7 The Hybrid algorithm stops when a predefined number of evolutions are reached otherwise go to step3.

In the hybrid GA-TS approach, TabuList is created and updated for better result that initializes best chromosome and then apply crossover and mutation.

A straightforward hybrid approach can be obtained by applying TS in a equivalent framework to a set of starting solutions in order to produce a set of good quality solutions; then apply GA to recombine the elements so generated. The entire process is repeatedly applied using the recombined solutions as starting ones.

6. Experiments and Evaluation

The tests were conducted on a machine that had Intel Pentium Dual CPU T2390 @ 1.87 GHz Processor and 2 GB of RAM memory and Windows 7 Operating System

CloudSim simulator has been used to simulate the cloud computing environment.

6.1 CloudSim Overview

Need of Simulator

Experimental assessment is too much of work and “costly” for computing researchers?

A more workable alternative is the use of simulation tools. These tools open up the opportunity of evaluating the hypothesis in a managed environment where one can effortlessly reproduce results. Simulation-based methods propose considerable benefits to IT companies (or anyone who wishes to offer his application services through clouds) by permitting them to: (i) experiment their services in repeatable and manageable environment; (ii) tune the system bottlenecks before hosting on real clouds; and (iii) trial with different workload mix and resource performance situations on simulated infrastructures for producing and testing adaptive application provisioning approaches.

Using CloudSim, academicians, researchers and industry-based developers can experiment the performance of a recently developed application service in a managed and easy to set-up environment. Based on assessment results reported by CloudSim, they can further enhance the service performance. The major benefit of using CloudSim for preliminary performance testing contain: (i) time efficiency: it needs very less attempt and time to apply Cloud-based application provisioning test environment and (ii) flexibility and applicability: researchers can model and check the performance of their application services.

CloudSim offers a comprehensive and extensible simulation framework that facilitates modeling, simulation, and testing of rising Cloud computing infrastructures and application.

Extensible CloudSim Components

A CloudSim component can be a class (abstract or complete), or set of classes that correspond to one CloudSim model (data center, host).

A Datacenter can control numerous hosts that in turn handle VMs during their life cycles

CloudSim helps the progress of custom application service models that can be hosted within a VM instance and its users are needed to expand the core Cloudlet object for implementing their application services. Once an application service is worked out and modeled, it is given to one or more pre-instantiated VMs through a service explicit allocation policy. Allocation of application-specific VMs to Hosts in a Cloud-based data center is the accountability of a Virtual Machine Allocation controller component (known as VmAllocationPolicy). The chief functionality of the VmAllocationPolicy is to select available host in a data center that meets the memory, storage, and availability requirement for a VM deployment. By default, VmAllocationPolicy employs a straightforward policy that allocates VMs to the Host in First-Come-First-Serve (FCFS) manner.

In this research work, by default FCFS VmAllocationPolicy has been replaced by GA, TS and Hybrid approaches and their results are also compared. Various hardware requirements such as the number of processing cores, memory and storage form the basis for such provisioning.

CloudSim supports numerous simulation circumstances that allocate specific CPU cores to particular VMs (a space-shared policy) or dynamically share out the capacity of a core among VMs (time-shared policy); and assign cores to VMs on demand.

Each Host component also instantiates a VM scheduler component, which can either, employ the space-shared or the time-shared policy for allocating cores to VMs.

CloudSim models scheduling of CPU resources at two levels: Host and VM.

At Host level, the host shares portions of each processor element (PE) to each VM executing on it. Because resources are shared among VMs, this scheduler is known as VmScheduler. Experimentation and implementation of our approaches has been done at this level.

In the VM level, each virtual machine fragments the resources received from the host among Cloudlets (In Cloud environment, tasks are called Cloudlets) running on it. Because in this level resources are shared among Cloudlets, this scheduler is known as CloudletScheduler. TimesharedCloudletScheduler policy has been used for the implementation in this research work.

Additional CloudSim components which are used in this work are discussed below.

DatacenterBroker: modifying the mode VM provisioning requests are submitted to data centers and the way cloudlets are submitted and assigned to VMs.

Cloudlet: This class models the Cloud-based application services.

Datacenter: It contents a set of computing physical machines or hosts that can either be homogeneous or heterogeneous with respect to their hardware configurations (memory, cores, capacity, and storage).

Host: This class represents a physical resource (or machine) such as a compute or storage server. It contents important information such as the amount of memory and storage, a list and type of processing cores, an allocation of policy for sharing the processing power among virtual machines, and policies for provisioning memory and bandwidth to the virtual machines.

Vm: This class forms a virtual machine, which is handled and hosted by a Cloud host component.

6.2 The QoS Metrics

In the cloud computing, QoS is a standard of user's fulfillment to the services. For example, some need more CPU time to work out complex task, and some others may require more memory to accumulate data, etc. Completion time, System cost, Resource utilization and Reliability are proposed as the QoS parameters for the research work when the job implement in the virtual machine resources.

- Completion time (or Deadline): For the real-time requirements of users, jobs need to be completed within a deadline.
- System Cost: Cloud computing is paid for demand. The cost is a factor which users concern.
- Resource utilization: Less number of resources should be utilized.
- Reliability: For the users' long-running tasks, cloud need to provide stable and reliable performance, such as cloud storage service.

According to expectations of QoS, parameter types can be defined. The smaller completion time, system cost and resource utilization, the better expectations of these parameters; the larger reliability, the better expectations of these parameters.

6.3 Test and Results

Numbers of experiments have been performed considering various QoS parameters using three approaches tabu search, genetic algorithm and hybrid algorithms. Their comparative results are also shown through different charts.

Experiment 1: Execution time comparison

Host is a CloudSim component that represents a physical computing server or machine in a Cloud: it is assigned a pre-configured processing capacity (expressed in millions of instructions per second – MIPS), memory, storage, and a provisioning policy for allocating processing cores to virtual machines. The Host component implements interfaces that support modeling and simulation of both single-core and multi-core nodes.

Every VM component has access to a component that stores the following characteristics related to a VM: accessible memory, processor, storage size.

Figures 5 (a and b) shows the execution time required to schedule number of VMs (Ranging from 75, 100, 125,150, 175, 200 … 500) to PMs (100, 300). It is observed that in most of cases, execution time for genetic algorithm is less as compared to TS and HA.

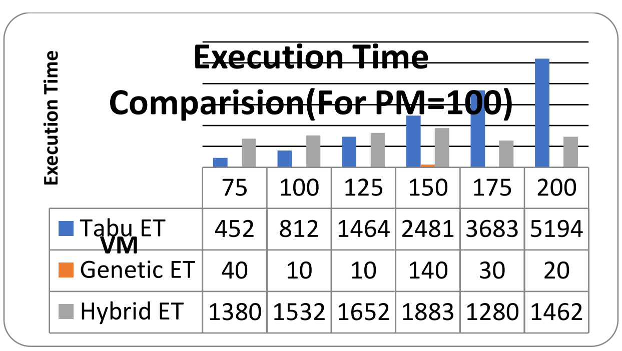
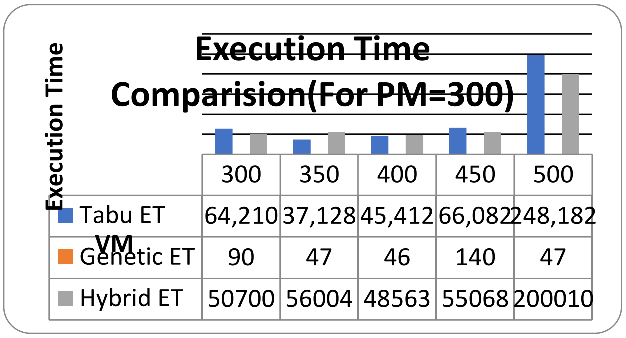


Figure 5(a): Execution Time comparison for PM = 100

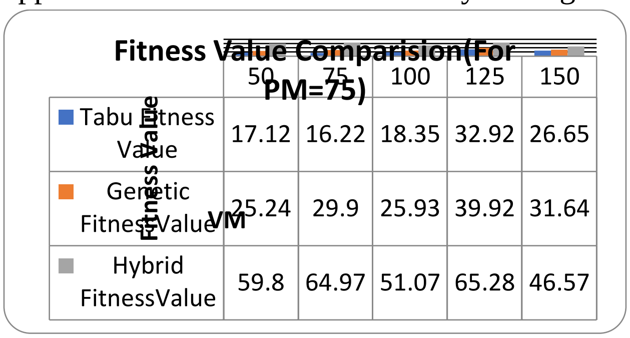


	300	350	400	450	500
Tabu ET	64,210	37,128	45,412	66,082	248,182
Genetic ET	90	47	46	140	47
Hybrid ET	50700	56004	48563	55068	200010

Figure 5(b) : Execution Time comparison for PM = 300

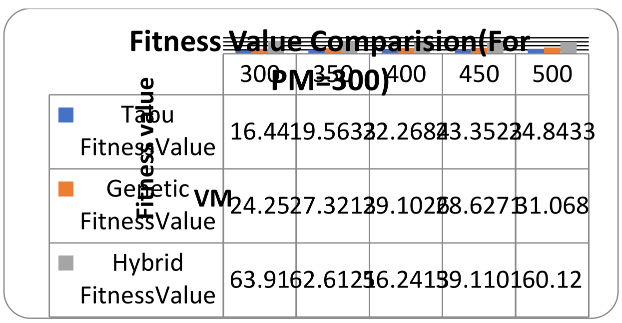
Experiment 2: Fitness Value Comparison

Better the fitness value, better the approach as fitness value determine the reliability that is needed to provide stable and reliable performance. Figure 6 (a and b) shows the fitness value comparison for three approaches and it is observed that hybrid algorithm has better fitness value in all the cases.



	50	75	100	125	150
Tabu Fitness Value	17.12	16.22	18.35	32.92	26.65
Genetic FitnessValue	25.24	29.9	25.93	39.92	31.64
Hybrid FitnessValue	59.8	64.97	51.07	65.28	46.57

Figure 6(a): Fitness Value Comparison (for PM = 75)



	300	350	400	450	500
Tabu FitnessValue	16.44	19.5633	32.2684	43.3523	34.8433
Genetic FitnessValue	24.25	27.3213	39.1026	68.6271	31.068
Hybrid FitnessValue	63.91	62.6125	56.2415	39.110	160.12

Figure 6(b): Fitness Value Comparison (for PM = 300)

Experiment 3: Resource Utilization Comparison

The main objective of an efficient scheduling approach is to properly utilize physical resources. Figure 7 (a and b) shows the comparisons among three approaches and in all instances hybrid algorithm has properly utilized the physical resources and has less physical machine count as compared to others approaches.

Resource Utilization Comparision(For PM=75)

VM	50	75	100	125	150
Tabu ResourceUtil	39	67	71	70	70
Genetic ResourceUtil	33	39	52	56	59
Hybrid ResourceUtil	21	37	37	48	49

Resource Utilization

Figure 7(a): Resource Utilization Comparison (for PM = 75)

Resource Utilization Comparision(For PM=300)

VM	300	350	400	450	500
Tabu ResourceUtil	253	258	270	271	282
Genetic ResourceUtil	163	190	204	217	232
Hybrid ResourceUtil	103	124	140	152	171

Resource Utilization

Figure 7(b):Resource Utilization Comparison (for PM = 300)

It is also observed that when 275 and 300 VMs are tried to be scheduled on 100 PMs, neither TS nor GA could schedule but hybrid algorithm did.

Experiment 4: System Cost Comparison

Cloud customers have to pay for the costs of memory and storage when they produce and instantiate VMs whereas the costs for network usage are only incurred in event of data transfer. Hence, if a Cloud customer provisions a VM without an application service (task unit), then they would only be charged for resources (i.e. the costs of memory and storage). Figure 8 (a and b) shows the simulation of different tasks on VMs using TimesharedCloudletScheduler policy. In most of the cases, system cost for hybrid algorithm is minimum.

System Cost Comparision(For PM=150, VM=150)

Tasks	150	175	200	225	250
TabuSystemCost	43.89	37.34	61.92	36.31	73.24
GeneticSystemCost	44.11	36.67	59.96	36.74	70.6
HybridSystemCost	43.72	36.98	58.4	36.28	68.84

System Cost

Figure 8(a): System Cost Comparison (for VM = 150)

System Cost Comparision(For PM=150, VM=225)

System Cost

Tasks

	225	250	275
TabuSystemCost	45.75	54.71	68.47
GeneticSystemCost	46.28	54.8	66.97
HybridSystemCost	45.82	54.41	66.91

Figure 8(b): System Cost Comparison (for VM = 225)

Experiment 5: Power Consumption Comparison

Cloud computing environments are constructed on an inter-connected network of a large number (hundreds-of-thousands) of computing and storage hosts for delivering on-demand services. Such infrastructures in combination with a cooling system, may consume massive amount of electrical power resulting in high operational costs. Lack of energy-conscious provisioning methods may lead to overheating of Cloud resources (compute and storage servers) in case of tremendous loads. This in turn, may result in reduced system consistency and lifetime of devices. So, power consumption is also an important factor when considering cloud computing environment.

Power Consumpsion Comparision(For PM=50, VM=100)

Power Consumpsion

Tasks

	50	100	125	150
Tabu PowerCons.	0.25	1.04	1.69	3.2
Genetic PowerCons	0.24	1.01	1.69	3.1
Hybrid PowerCons	0.24	1.03	1.67	3.07

Figure 9: Power Consumption Comparison

Experiment 6: Completion Time Comparison

For the real-time requirements of users, tasks need to be completed within a deadline. Figure 10 (a and b) shows the completion time of different tasks on different VMs and observed that in most of the cases hybrid algorithm produces minimum completion time.

Completion Time Comparision(For PM=50, VM=100)

Completion Time

Tasks

	50	100	125	150
Tabu CompletionTime	2.37	5	6.46	9.8
Genetic CompletionTime	2.28	4.83	6.41	9.6

Figure 10(a): Completion Time Comparison (for VM = 100)

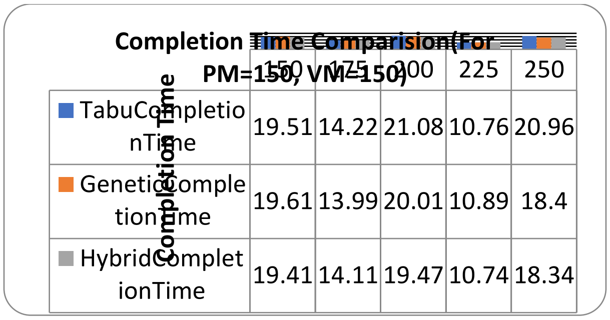


Figure 10(b): Completion Time Comparison for VM = 150)

7. Conclusions

This research paper attempted to address the hybridization of the two of the popular nature's heuristics namely GA and TS for task scheduling on Cloud Computing environment using CloudSim simulator. While GAs deal with population of solutions, TS is search procedures that handles only one solution at a time.

Global optimization algorithms attract significant computational effort. In a cloud environment, the main emphasize to generate the schedules at a minimal amount of time. Especially as the demand increases, when the number of tasks and the resources starts towering up, conventional heuristics algorithms become time consuming. Considering various QoS parameters, various experiments are performed and it is observed that hybrid algorithm performed much better than genetic algorithm and tabu search.

8. References

[1] Brian Hayes, Cloud computing, Communications of the ACM, Volume 51, Issue 7, July 2008.

[2] R. Buyya, C. S. Yeo, S. Venugopal, “Market-oriented Cloud Computing: Vision, hype, and reality for delivering it services as computing utilities,” CoRR, vol. abs/0808.3558, 2008.

[3] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, ‘‘Above the Clouds: A Berkeley view of Cloud Computing,’’ Univ. of California, Berkeley, Tech. Rep., 2009.

[4] Hamscher, V., et al., Evaluation of Job-Scheduling Strategies for Grid Computing. LECTURE NOTES IN COMPUTER SCIENCE, 2000: pp. 191-202.

[5] Zhiguang, S. and L. Chuang. Modeling and Performance Evaluation of Hierarchical Job Scheduling on the Grids. in Grid and Cooperative Computing, 2007. GCC 2007. Sixth International Conference on. 2007.

[6] Bayati, M., et al. Iterative Scheduling Algorithms. in INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE. 2007.

[7] Patel, Y. and J. Darlington. A Novel Stochastic Algorithm for Scheduling QoS-Constrained Workflows in a Web Service-Oriented Grid. in Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on. 2006.

[8] Carona, E., V. Garonneb, and A. Tsaregorodtsevc, Definition, modelling and simulation of a grid computing scheduling system for high throughput computing Future Generation Computer Systems, 2007. Volume 23(Issue 8): p. 968-976.
[9] Weidong, H., Y. Yang, and L. Chuang. Qos Performance Analysis for Grid Services Dynamic Scheduling System. in Wireless Communications, Networking and Mobile Computing, 2007. WiCom 2007. International Conference on. 2007.
[10] Afzal, A., A.S. McGough, and J. Darlington, Capacity planning and scheduling in Grid computing environments. Future Generation Computer Systems 2008. : p. 404-414.
[11] Kiran, M., et al. A prediction module to optimize scheduling in a grid computing environment. in Computer and Communication Engineering, ICCCE International Conference on. 2008.
[12] Rasooli, A., M. Mirza-Aghatabar, and S. Khorsandi. Introduction of Novel Rule Based Algorithms for Scheduling in Grid Computing Systems. in Modeling & Simulation, AICMS Second Asia International Conference on. 2008.
[13] Yuan-Shun, D., X. Min, and P. Kim-Leng, Availability Modeling and Cost Optimization for the Grid Job Management System. Systems, Man and Cybernetics, Part A, IEEE Transactions on, 2008. 38(1): p. 170-179.
[14] S. Venugopal, X. Chu, and R. Buyya. ‘‘A Negotiation Mechanism for Advance Job Reservation using the Alternate Offers Protocol.’’ Proc. of IEEE IWQoS, June 2008
[15] K. Xiong and H. Perros. ‘‘SLA-based service composition in enterprise computing’’. Proc. of IEEE IWQoS, pp 30-39, 2008.
[16] D. Ardagna, M. Trubian, and L. Zhang. ‘‘SLA based job allocation policies in autonomic environments’’. Journal of Parallel and Distributed Computing, vol.67, pp.259--270, 2007.
[17] Luqun Li, “ An Optimistic Differentiated Service Job Scheduling System for Cloud Computing Service Users and Prividers”, Third International Conference on Multimedia and Ubiquitous Engineering, 2009: pp 295-299
[18] Jiayin Li, Meikang Qiu, Jianwei Niu Wenzhong Gao Ziliang Zong Xiao Qin, “ Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems”, IEEE/WIC/ACM International Conference on Intelligence and Intelligent Agent Technology, 2010: pp 561-564.
[19] Masnida Hussin, Young Choon Lee, Albert Y. Zomaya,”Dynamic Job-Clustering with Different Computing Priorities for Computational Job Alloction”, 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010: pp 589-590.
[20] H. Izakian, A. Abraham, and B. T. Ladani, “An Auction method for job allocation in computational grids,” Future Generation Computer Systems, vol. 26, no. 2, pp. 228-235, 2010.
[21] M. Higashida. High Performance Computing on Vector Systems 2009, chapter The Grid Middleware on SX and Its Operation for Nation-Wide Service, pages 109–119. Springer Berlin Heidelberg, 2009.
[22] Jaliya Ekanayake, Thilina Gunarathne, Judy Qiu, “Cloud Technologies for Bioinformatics Applications”, IEEE Transactions On Parallel And Distributed Systems, Vol. 22, No. 6, June 2011
[23] Di-Wei Huang, Jimmy Lin,” Scaling Populations of a Genetic Algorithm for Job Shop Scheduling Problems using MapReduce”, 2nd IEEE International Conference on Cloud Computing Technology and Science, 2010.

[24] Attila Csaba Marosi and P´eter Kacsuk,” Workers in the Clouds”, 19th International Euromicro Conference on Parallel, Distributed and Network-Based Processing, 2011: pp 519-524
[25] OpenStack Open Source Cloud Computing Software, http://openstack.org. Cited 08 July 2010
[26] Condor Workers on Amazon EC2: http://www-rcf.usc.edu/_juve/condor-ec2/, Cited 08 July 2010
[27] Wei, Zhang-liang Li, Yue-guang,” An ant colony algorithm with tabu search and its application”, Fourth International Conference on Intelligent Computation Technology and Automation, 2011: pp 412-416.
[28] L. Zhao, Y. Ren, Y. Xiang et al. Fault tolerant scheduling with dynamic number of replicas in heterogeneous system. 12th IEEE International Conference on High Performance Computing and Communications, HPCC2010, Melbourne, 2010.
[29] Prasad and Faruquie and Subramaniam and Mohania,”Job Allocation and SLA Determination for Large Data Processing Services Over Cloud.” IEEE ICSC, 2010.
[30] Bossche and Vanmechelen and Broeckhove, ”Cost-Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads.” IEEE IC on Cloud Computing, 2010.
[31] “Performance evaluation of bag of gangs scheduling in a heterogeneous distributed system,” The J. of Syst. and Software, pp. 1–9, 2010.
[32] J. Zelenka, “Parallel computing application into the Particle Swarm Optimization algorithm used to solve the Job-Shop Scheduling problem”, 15th International Conference on Intelligent Engineering Systems, June 23–25, 2011, Poprad, Slovakia : pp 183-189.
[33] Zelenka, J., Kasanický, T.: Comparison of artificial immune systems with the particle swarm optimization in job-shop scheduling problem. In SAMI 2011: 9th IEEE International Symposium on Applied Machine Intelligence and Informatics. -Budapest : IEEE, 2011, p. 129-135. ISBN 978-1-4244-7428-8.
[34] Shiau, D., “A hybrid particle swarm optimization for a university course scheduling problem with flexible preferences.” Expert Systems with Applications, 38, pp. 235-248, 2011.
[35] D. P. Lin and C. K. M. Lee, "A review of the research methodology for the re-entrant scheduling problem," International Journal of Production Research, 2010.
[36] Afshin Kazerooni, Reza Ebrahimpour, Morteza Saedi Dezaki,” Single machine scheduling problem of minimizing maximum earliness and number of tardy jobs using a genetic algorithm”, International Conference of Soft Computing and Pattern Recognition (SoCPaR),2011 : pp 402-406
[37] Sivadon Chaisiri, Bu-Sung Lee, Dusit Niyato,” Optimization of Job Provisioning Cost in Cloud Computing”, IEEE TRANSACTIONS ON SERVICES COMPUTING, January 31, 2011.
[38] Jiahui Jin, Junzhou Luo, Aibo Song, Fang Dong, Runqun Xiong, “BAR: An Efficient Data Locality Driven Task Scheduling Algorithm for Cloud Computing”, 11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing,2011 : pp 295-304
[39] D. Jiang et al., “The performance of mapreduce: An indepth study,” PVLDB, vol. 3, no. 1, pp. 472–483, 2010.
[40] Yanbin Liu, Norman Bobroff, Liana Fong, Seetharami Seelam, Javier Delgado,” New Metrics for Scheduling Jobs on a Cluster of Virtual Machines”, IEEE International Parallel & Distributed Processing Symposium 2011: pp 1001-1005.

[41] Jong Myoung Ko, Chang Ouk Kim, Ick-Hyun Kwon," Quality-of-service oriented Web service composition Algorithm and planning architecture, The Journal of Systems and Software Vol 81, pp.2079-2090, 2008.
[42] Mrs.S.Selvarani, Dr.G.Sudha Sadhasivam," Improved Job-Grouping Based Pso Algorithm For Task Scheduling In Grid Computing", International Journal of Engineering Science and Technology Vol. 2(9), 2010, 4687-4695
[43] Hengliang Shi, Guangyi Bai, Zhenmin Tang," ACO Algorithm-based Parallel Job Scheduling Investigation on Hadoop", International Journal of Digital Content Technology and its Applications. Volume 5, Number 7, July 2011
[44] I. Moschakis and H. Karatza, "Evaluation of gang scheduling performance and cost in a cloud computing system," The J. of Supercomput., pp. 1–18, 2010.
[45] Shiyao Chen, Ting He, Ho Yin Starsky Wong, Kang-Won Lee, Lang Tong," Secondary Job Scheduling in the Cloud with Deadlines", IEEE International Parallel & Distributed Processing Symposium,2011: pp 1009-1016
[46] Seonho Kim and Jon B. Weissman, "A GA-based Approach for Scheduling Decomposable Data Grid Applications"
[47] D. Bogdanovich, A. Ziyangirov, "Genetic algorithm structure optimization for the schedule task solution", IEEE 2011
[48] GAN Guo-ning, HUANG Ting-Iei, GAO Shuai," Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment" IEEE 2010, pp 60-63
[49] HUANG Qi-yi, HUANG Ting-lei," An Optimistic Job Scheduling Strategy based on QoS for Cloud Computing" IEEE 2010 pp 673-675
[50] Yujia Ge, Guiyi Wei," GA-Based Task Scheduler for the Cloud Computing Systems" 2010 International Conference on Web Information Systems and Mining, pp 183-186
[51] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu," Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing" 2009 IEEE
[52] Dominique A. Heger," Optimized Job Allocation & Task Scheduling Challenges in Cloud Computing Environments"
[53] S. Meeran, M. S. Morshed, "A hybrid genetic tabu search algorithm for solving job shop scheduling problems: a case study", Journal of Intelligent Manufacturing August 2012, Volume 23, Issue 4, pp 1063-1078
[54] Zhongyuan Lee, Ying Wang, Wen Zhou," A dynamic priority scheduling algorithm on service request scheduling in cloud Computing", 2011 International Conference on Electronic & Mechanical Engineering and Information Technology,pp 4665-4669
[55] Bajeh, A. O., Abolarinwa, K. O.," Optimization: A Comparative Study of Genetic and Tabu Search Algorithms", International Journal of Computer Applications (0975 – 8887) Volume 31– No.5, October 2011, pp 43-48
[56] Arash Ghorbannia Delavar, Yalda Aryan," Synthetic Heuristic Algorithm for Independent task Scheduling in Cloud Systems", IJCSI International Journal of Computer Science Issues, Vol. 8,

Issue 6, No 2, November 2011, pp 289-295
57] Lizheng Guo, Shuguang Zhao1, Shigen Shen1, Changyuan Jiang," Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm", JOURNAL OF NETWORKS, VOL. 7, NO. 3, MARCH 2012 pp 547-553
[58] V. Krishna Reddy and Dr. L.S.S. Reddy," A survey of various task scheduling algorithms in cloud environment", GJCAT, Vol 2 (1), 2012, pp 847-853
[59] Suresh.A, Vijayakarthick.P," Improving scheduling of backfill algorithms using balanced spiral method for cloud metascheduler", IEEE-International Conference on Recent Trends in Information Technology, ICRTIT 2011,pp 624-627
[60] Zhongni Zheng ,Rui Wang, Hai Zhong, Xuejie Zhang," An Approach for Cloud Job Scheduling Based on Parallel Genetic Algorithm", 2011 IEEE, pp 244-247
[61] Joanna Kołodziej, Fatos Xhafa," Modern Approaches To Modeling User Requirements On Job And Task Allocation In Hierarchical Computational Grids", Int. J. Appl. Math. Comput. Sci., 2011, Vol. 21, No. 2, 243–257
[62] Ajith Abraham, Rajkumar Buyya and Baikunth Nath," Nature's Heuristics for Scheduling Jobs on Computational Grids"
[63] Salim Bitam," Bees Life Algorithm for Job Scheduling in Cloud Computing",ICCIT 2012, pp 186-191
[64] Jianhua Gu, Jinhua Hu, Tianhai Zhao, Guofei Sun," A New Job Scheduling Strategy Based on Genetic Algorithm in Cloud Computing Environment", JOURNAL OF COMPUTERS, VOL. 7, NO. 1, JANUARY 2012,pp42-52
[65] Mohammad Ranjbar & Mojtaba Najafian Razavi, "A hybrid metaheuristic for concurrent layout and scheduling problem in a job shop environment", Int J Adv Manuf Technol (2012) 62:1249–1260
[66] Fardin Ahmadizar & Mehdi Hosseinabadi Farahani, "A novel hybrid genetic algorithm for the open shop scheduling problem", Int J Adv Manuf Technol (2012) 62:775–787
[67] Hong-an Yang, Qi-feng Sun, Can Saygin & Shu-dong Sun," Job shop scheduling based on earliness and tardiness penalties with due dates and deadline: an enhanced genetic algorithm", Int J Adv Manuf Technol(2012) 61:657-666
[68] Thamilselvan, R. and P. Balasubramanie, "Integration of Genetic Algorithm with Tabu Search for Job Shop Scheduling with Unordered Subsequence Exchange Crossover", Journal of Computer Science 8 (5): 681-693, 2012
[69] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms", Journal of Software: Practice and Experience, Volume 41, Issue 1, Pages 23-50, January 2011.
[70] Sachin V. Solanki, Dr. (Mrs.) A. R. Mahajan "An Optimized Strategy Based on Genetic Algorithm for Predicting Best Candidate" 2013 Elsevier Proceedings of International Conference on ComNet CIIT & ITC organized by the Association of Computer Electronics and Electrical Engineers (ACEEE), pp 794-798, ISBN 978-81-910691-6-3.

[71] Sachin V. Solanki, Dr. (Mrs.) A. R. Mahajan "Job Scheduling based on the implementation of Genetic Algorithm" the International Conference on Information and Communication Technology for Competitive Strategies (ICTCS - 2014) organized by Computer Society of India Udaipur Chapter and conference proceedings published by ACM-ICPS Proceeding Volume ISBN No. 978-1-4503-3216-3, http://dx.doi.org/10.1145/2677855.2677904.