Comparison of Probabilistic Optimization Algorithms for Resource Scheduling in Cloud Computing Environment

Mayank Singh Rana [*1], Sendhil Kumar KS [*2], Jaisankar N [*3]
[*] School of Computing Science and Engineering, VIT University
Vellore, Tamil, Nadu, India – 632 014
[*1] mayanksingh.rana@yahoo.com
[*2] sendhilkumar.ks@vit.ac.in
[*3] njaisankar@vit.ac.in

Abstract—**Cloud Computing has significantly made its landmark in the field of information technology. A concept which initially stood nebulous is now used as a synonym for internet. It has paved a way to increase capacity as well as add capabilities on the fly without investing in new infrastructure, training new personnel, or licensing new software."Cloud computing in broad sense can be understood as a connected pool of computing resources (including software and hardware) which can eventually be delivered as a service over a network (in broad sense the Internet)". Resources are available to the user in utility-style infrastructure. However management of resources in a dynamic fashion stands a vast area for researchers. These resources are precisely available at certain fixed times and also for fixed intervals of time. Thus Scheduling of these resources constitutes a major part of resource management. An optimized scheduling of resources is required to maintain separation between users of the resources. In this paper, we discuss various resource scheduling strategies that have been and are being implemented in cloud computing environments. We also propose a comparison among the characteristics features of these resource scheduling algorithms.**

Keyword-Ant Colony, Genetic Algorithm, Optimization Techniques, Resource Scheduling, Swarm Intelligence.

I. INTRODUCTION

Cloud computing has become a quite-essential part of many developed and also developing firms/organizations. The major potential of the cloud comes from its ability to provide flexible and on the run resources, in terms of services. Cloud computing is also claimed the title of specialized computing due to the presence of a unique feature of the cloud computing called utility computing. The efficiency of cloud computing environment greatly relies on its ability to not only maintain the resources but also on effectively scheduling those resources. Resource scheduling inherently contributes to the strength of the cloud computing. Just like all other scheduling algorithms, resource scheduling algorithms also focuses on optimal usage of the available resources. In addition to that the primary focus of these algorithms is to schedule the resources to in a manner that satisfies the cloud user's to a maximum possible extent. Cloud computing resources spans over a vast range; including infrastructure, software, storage, security, data, etc which are delivered to the user in the form of pay-per-use services. Adequate resource scheduling not only helps the cloud provider but benefits each person/group associated with the cloud.

Proper resource scheduling benefits the cloud provider in terms of efficient resource management which in turn provides more resources to allocate without postponing or declining any user requests. On the other hand, cloud users are benefitted in terms of their monetary gains at each front. Optimizing resource scheduling techniques helps elevate these gains in an efficacious manner. Some of the prominent resource scheduling algorithms comprise of the Ant Colony algorithm, Particle swarm technique, genetic algorithm etc. These algorithms are primarily optimization algorithms and they are being implemented in the cloud computing environment to help optimization of resource scheduling. These algorithms are now being used for quite some time for scheduling of resources, however over the span of time each one of them have been altered and modified in order to make them more and more efficient and stringent towards their goal. In this paper, we present a review of various optimization algorithms used for scheduling of resources in cloud and also their evolution over the period of time.

II. OPTIMIZATION ALGORITHMS

One of the fundamental principles for every object in the universe is to stand in the best possible optimal state. "In almost all computing and non-computing fields and also in mathematics, statistics, empirical sciences, or even management science, in nearly every field optimization plays the most essential role. Optimization can briefly be defined as, the selection of best element (pertaining to specified criteria) from an available set of variable alternatives". Optimization can be achieved in various forms and by applying numerous methods; however the goal of each one is to achieve – "maximal output with minimal input".

Optimization can be classified into wide range of categories based on the type of output desired. It can be a) the need to arrive on a feasible point b) to detect the existence of a particular problem/situation c) to explore the necessary and sufficient conditions etc. However the cloud computing environment requires optimization algorithms to optimize the resource usage by providing convenient/optimal schedule. Optimization is widely categorized into 3 broad fields, deterministic, probabilistic and heuristic algorithms. In this paper our primary concern is on the probabilistic algorithms. In cloud environment probabilistic algorithms are employed for efficient resource usage. [24] The probabilistic algorithm family involves an especial family of algorithms named 'Monte-Carlo algorithms'. Out of the various algorithms in Monte-Carlo the most prominent algorithms used in cloud are Ant Colony, Particle Swarm and Genetic algorithms. Though there had been other scheduling algorithms, however they rendered themselves unfruitful in the due course of time. We will discuss the most widely implemented algorithms in the following section.

III. ANT COLONY ALGORITHM

"The ant colony optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding best available paths through graphs.". Ant Colony Algorithm is essentially a category of swarm intelligence. [5]. Ant Colony algorithm is based on the characteristic feature of an ant. Ants, in spite of their absence of eyes, could easily find their food and also trace back there path to their homes. Ants use 'pheromone deposition' in order to accomplish these tasks. Initially they deposit pheromones on the ground to mark the path used by them to discover the way to the food. Secondly, they deposit the pheromone indicating the optimized route between the food source and the home.

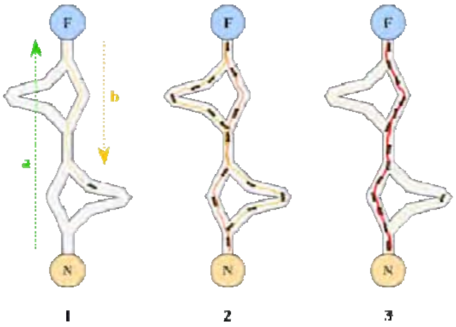

Fig. 1. Bridge experiment depicting the pheromone principle for Ant Colony Algorithm

Let us discuss where this algorithm does finds its application and how does it work. Consider a fully functional cloud computing environment; A host/cloud provider with various resources and virtualised distribution of resources on different Virtual Machine's (VM's). On the other hand we have many cloud users connected to the cloud provider on the pay-as-you-go basis, as shown in fig 2. The end user can be any single person, group of individuals, a company, an organisation, of other giant firms. Now each cloud user submits its job to the cloud provider specifying SLA (Service Level Objectives) which includes resource requirements, time to accomplish the task, etc and other relevant details. The user pays to the cloud provider while he is accessing the resources allocated to him by the cloud provider. Similarly all the cloud users submit their tasks to the cloud provider specifying all the essential details. The cloud provider starts by gathering all the requests (in a particular time stamp) and then allocating suitable resources to each of them. At this point Ant Colony algorithm plays a helpful role. The cloud provider needs to allocate jobs to various Virtual Machines. Job scheduling is of vital importance in this scenario. Here all the virtual machines are analogous to the nodes, ants are analogous to the mobile agent, and resources are similar to the food source in case of ant colony. Ant traverses amongst the nodes to allocate the jobs to various resources in an optimized manner using Ant Colony Algorithm.

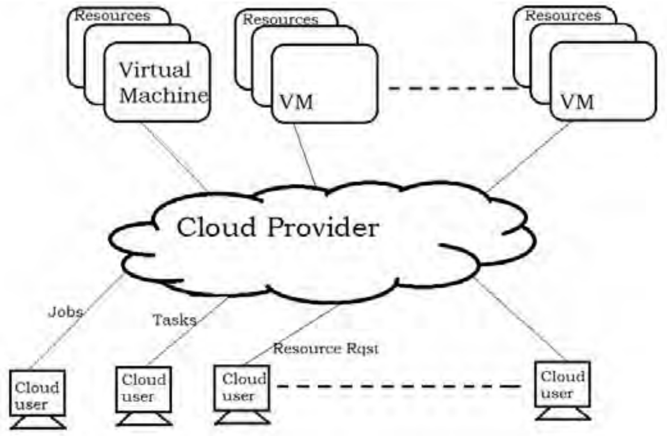

Fig. 2. Architecture of basic cloud resource allocation schema

In this scenario each node would be configured with two important aspects: I) Probability of being a destination II) Maintains a pheromone (or probabilistic routing) table [17]. The table at each node is configured in such a way that the rows represent routing preference for each destination while the columns would represent the probability of neighbour being the next node. Thus ants travelling from each node will update the entries in the table at each node [17]. Applying Ant Colony Algorithm at various nodes in cloud provides optimized scheduling of jobs which in turn result in efficient Resource Management.

IV. OPTIMIZED ANT COLONY ALGORITHM

Ant colony algorithm though being an efficient way for resource scheduling does suffer from some of inherent drawbacks. Like the table updating strategy may sound perfect however in case of asymmetric network of nodes it may fail to deliver best results in cloud environment. Also in the case where more VM’s are involved the ant colony algorithm would develop colonies, so an ant is less likely to follow a pheromone trail from another colony.

(i) In order to avoid such short comings there were several reforms were made to the traditional Ant colony algorithm to achieve an optimized version of Ant Colony algorithm. Over the period of time there were several changes made to the ant colony algorithm. The major ones are listed below:

(ii) Creating gateways for each set of VM’s; each group of mobile agents corresponds to a colony of ants, and the routing table of each group corresponds to a pheromone table of each colony [17]. In this way the ant/mobile agent in spite of having its own routing preferences will take into consideration the one encountered.

(iii) The mobile agents/ants will automatically update the information from local knowledge table to Global knowledge table [18].

(iv) In order to reduce time taken to execute resource allocation requests, while updating the tables load on each VM is also taken into consideration [11].

(v) Classifying the pheromone for the forward and backward movements. As in backward movement; when an ant encounters an overloaded node after it has encountered an overloaded node then it travels in the backward direction leaving a trailing pheromone. And vice versa in case of forward movement leaving a foraging pheromone [12].

V. PARTICLE SWARM OPTIMIZATION

It is a computational method which belongs to the super set of swarm intelligence. In this algorithm iterative procedure is used to optimize a problem. The candidate solutions are iteratively processed with respect to a given measure of quality. In PSO, population of candidate solutions is referred to as ‘particles’. The optimization in PSO involves movement of these particles around in the search-space. The movement is guided by simple mathematical formulae over the particle's velocity and position. Each particle’s local best known position influences its movement. In addition to this it is also guided toward the best known positions in the search-space, which are updated as better positions are found by other particles. The swarm is supposed to converge towards the optimal solution by these particle movements.

Like Ant colony algorithm, Particle swarm algorithm is also a Metaheuristic algorithm. A simple form of algorithm works by starting a population of the candidate solutions (the Swarm). Each candidate solution is analogous to a particle in the system. The system is supposed to be the solution space. These particles are then moved around in the search-space on basis of some mathematical formulae with the aim to converge to an optimum solution.

Consider a simple cloud computing environment where various users are submitting their jobs with request for specific resources to the cloud provider. Now at any time instant the cloud provider need to schedule the jobs to the available resources in the best possible manner. In this scenario, Particle Swarm Algorithm serves to be of great use. Each mobile agent in the cloud environment could serve as a potential candidate or so called particle. [23]The particle moves with a specific velocity with defined magnitude and in defined direction. This movement is associated with the best solution attained so far. Also a fitness value is used to indicate the performance of the particle. It keeps a track of each position at each time instant, as shown in Fig. 3 and Fig. 4. The movement is guided towards attaining a better solution by varying the acceleration at each time step. Ultimately when the particle covers the entire search space making them its topological neighbor the final best solution is generated, as shown in Fig. 4 [20].

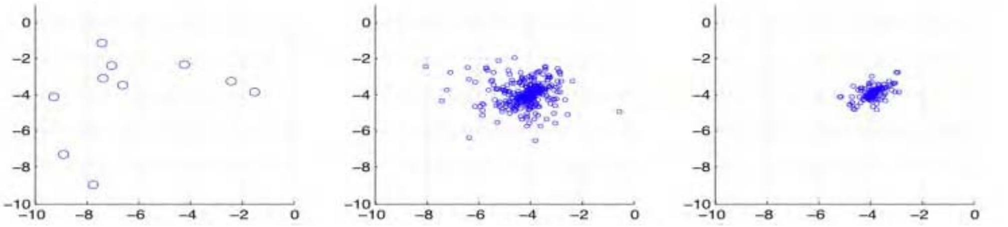

Fig. 1. The movement of particles to attain the best solution

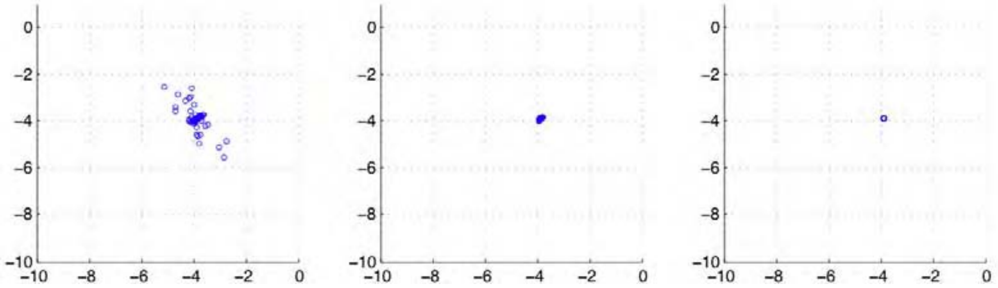

Fig. 2. Attaining the best solution after varying the acceleration at each time step

VI. OPTIMIZED PARTICLE SWARM ALGORITHM

As said Particle Swarm Algorithm is a metaheuristic approach, thus it needs to make certain assumptions to attain the best solution. Also the solution space or the search space can end up being very large. Metaheuristics such as PS may not even be able to guarantee an optimal solution is ever found. In terms of mathematical form also PSA doesn’t considers the gradient of the problem being optimized which means that problem may not be differentiable. However over the years of research done in the field of PSA some elementary modifications are made to the algorithm, to make it more efficacious.

Some of them are listed below:-

(i) Dynamic Hierarchical PSO

In this the next population, after the initial start is allowed to explore the patent of the earlier population. This gives an advantage because it is no longer required to predetermine the no. of clusters to be found in the solution space[9].

(ii) Revised Discrete PSO

It emphasises on calculating more accurately the position and velocity of the particle. This is done by defining some operation rules and also defining their equation of motion in accordance with few discrete variables[26].

(iii) Modified PSO

In this method the rate at which the position of the particle changes is given by a modified velocity equation an mentioned below [25]:

$$V_{id} = V_{id} + c_1 * rand() * (p_{id} - x_{id}) + c_2 * Rand() * (p_{gd} - x_{id})$$

Where $x_{id} = x_{id} + v_{id}$

VII. GENETIC ALGORITHM

In the field of Artificial Intelligence, the algorithm working of search heuristics and mimics the natural process of evolution is profoundly termed as Genetic Algorithm. As the name genetic algorithm works widely in a similar fashion to that of natural evolution. Using continual search heuristics, the initial solution is modified over time in order to gain the closest optimal solution as desired of problem. The process goes as; initially to begin with we have a population of strings. These are also referred as the genome or even genotype in order to provide alias with the algorithm. These populations of strings or (commonly known as chromosomes) are used to encode the candidate solutions for the problem. The candidate solutions are also termed as phenotypes on individual creatures. The candidate solutions encoded in the population string is evolved towards a more desired solution.

Usually the population is encoded in a binary form using 0 and 1, however other types of encoding are also possible. The evolution begins with population of solutions that are randomly generated and goes on towards a better generation. Whenever a new generation is obtained, its fitness is checked for each individual in it. After this multiple individuals are selected which are most suitable (in terms of fitness) in the current population which are then modified. Modification is done by combining (mutation) various individuals to generate a new population of modified individual. This generation of new population is again subjected to the same process, iteratively. After attaining a satisfactory level the algorithm terminates. However if the algorithm terminates because of large no. of populations, then the satisfactory level of population might not be achieved. [2] Thus, the entire working of Genetic algorithm can be summarized into 5 major steps:-

(i) Initialisation
(ii) Selection
(iii) Cross-Over
(iv) Mutation
(v) Termination

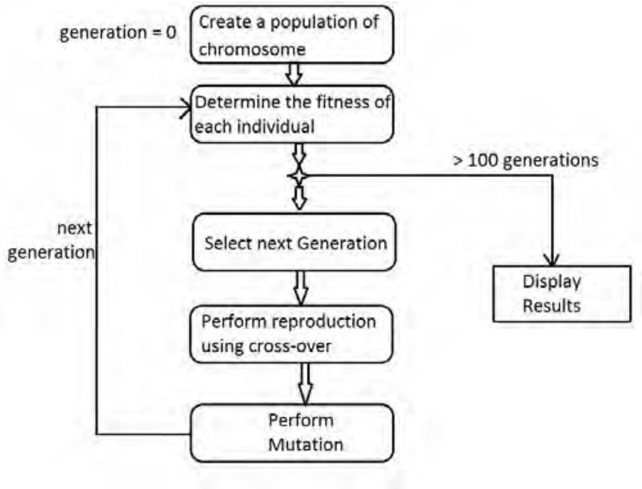
generation = 0
Create a population of chromosome
Determine the fitness of each individual
> 100 generations
next generation
Select next Generation
Display Results
Perform reproduction using cross-over
Perform Mutation

Fig. 5: Flowchart of Genetic Algorithm

Cloud computing, which stands for providing resources to all users in the form of service, must have a fast and unique methods to understand the needs of the user and allocate them the resources as soon as possible. Genetic algorithm being the most effective in evolution seems to fit perfect for allocating the resources to users and also optimize the resource usage with appropriate scheduling [19].

VIII. OPTIMIZED GENETIC ALGORITHM

Genetic algorithm has been an area of wide interest for many researchers. Its evolutionary concept has been studied and being modified to suit the need of the user as well as the resource providers. Some of the most prominent modifications made in genetic algorithm are as follows:-

(i) Robust Genetic Algorithm for resource scheduling

In this method the encoding stage is further explored to consist of the following: Activity-List representation, Priority rule representation, Random Key representation, Shift Key representation. These perform in the effective way to reach an optimal solution [10].

(ii) Genetic Algorithm using DSS

Using DSS (Decision-Support-System) it was found that it can provide optimal solutions or atleast near-optimal solutions in most of the problems [22].

(iii) Modified Genetic Algorith

Matthew Bartschi Wall proposed in his work [15] that using modified Genetic algorithm not only provides flexibility but also makes the process more robust. This method used a dual array for encoding purpose which unlike the traditional method is direct and time-based representation.

(iv) Breeder Genetic Algorithm

The selection phase in this method is inspired by the selection used in human breeding. Its model is derived from quantitative genetics. It compares behavior of different mutation schemes, which gives the progress to the desired optimal solution. [7]

(v) Fuzzy Genetic Algorithm

The central idea of this algorithm is to give a choice to the core/central scheduler. This choice is in reference of the global view of the entire system [21].

(vi) Improved Genetic Algorithm

This method is guided by the sole principle of maximum utilization of resources by effective allocation of more and more VM's. It works by using the idea of shortest genes. It also introduced the concept of Dividend Policy in order to choose an optimal or nearly optimal allocation for VMs [8].

IX. COMPARATIVE ANALYSIS

In the previous sections, the basic working principle, characteristics and the evolution of a few scheduling algorithms has been discussed. In this section, a tabular comparison of these scheduling algorithms is presented in order to analyse their performance and features. A linear mapping of each of the algorithms important characteristics is shown in the table below.

TABLE I
Comparative analysis of various Resource optimization algorithms

	ANT Colony Optimization	**Particle Swarm Optimization**	**Genetic Algorithm**
Categorization	It belongs to the class of Swarm Intelligence	It belongs to the class of Swarm Intelligence	It belongs to the class of Evolutionary Algorithms
Inspiration	Inspired by the behavior of ants in their local proximity, to search food or to seek path back to home	Inspired by the movement of birds in flocks or fishes in school	Inspired by the higher class of natural evolution like inheritance
Initialization	Initialized with 'ants' representing the initial state of the problem	Initialized with 'particles' representing the initial state of the problem	Initializes with population of strings encoded in the candidate solution
Core concept	Collective behavior of self-organized or decentralized system which may be natural or artificial	Collective behavior of self-organized or decentralized system which may be natural or artificial	Governed by the natural laws of evolution
Technique	It is a probabilistic technique employed to handle computational tasks by finding minimized route using graphs	It is a computational method which works iteratively on candidate solution in order to optimize a problem with respect to certain parameters	It serves as the basis for artificial intelligence where search heuristics are employed which mimics the process of natural evolution
Working Principle	Pheromone trails from ants	Varying the speed with which the particle moves in the search space	Survival of the fittest in natural evolution
Methodology	The pheromone trail from ants contains the information about the optimized route. The thicker the trail more no. of ants would follow whenever crosses it.	The particle is moved in the search space with varying intensity guided by parameters and its position is measured at each time instant.	After initializing, selected chromosomes go through mutation. Later the fittest are carried forward while others are terminated.

X. TOOLS FOR CLOUD SIMULATION

Cloud has become very popular because of its ability to satisfy the needs in the best possible manner. This recent technology has attracted more and more to researchers towards itself. The recent technological advancements not only concerns developing the most efficient resource scheduling algorithm but also considerable portion of researchers are working on developing the most efficient cloud computing simulation tool. There are various tools available in the market for cloud simulation. Some of these tools are services which are potentially capable to provide all types of services. Some of the most prominent tools for cloud computing environment simulation are discussed below:-

A. Eucalyptus (Latest Release: Eucalyptus 3.2.1)

It is an open source cloud computing system. Eucalyptus provides a framework for software systems in cloud computing environment, especially for IaaS (Infrastructure as a Service). Eucalyptus not only provides some software as a service, it provides an entire virtual computing environment to the user as a service. It consists of a very simple architecture, in which all components interact with each other in a hierarchical manner. It provides user a familiar interface, which enables them to use tools in the same manner as they would be using on Amazons EC2. Operating systems and hypervisors for Eucalyptus are:-

a) All versions of Ubuntu

b) CentOS

c) RHEL

d) VMWare (All the above with x86_64 architecture)

B. CloudSim (Latest Version : CloudSim Toolkit 3.0)

This stands for CLOUD SIMulation, is a framework for modelling as well as simulating various services and infrastructures on cloud. CloudSim can provide virtualised environment to users even on the fly while maintain the quality of service provided to the end user. It leverages virtualised environments based on the user requirement such as workload which varies with time. It provides user both the facility to either model the cloud or simulate various services to the intended user. Another attractive feature of CloudSim is that, it provides user with a custom interface which allows user to decide implementing policies for the VM’s. It provides nearly all types of major services like IaaS, PaaS and SaaS. Some of its main features are illustrated below:-

Modelling and simulation support for

a) Virtualized servers

b) Resources with energy aware computing

c) Topology with data centred network

d) Cloud federation[3]

C. Aneka (Latest Version : Aneka 3.0)

For developing cloud applications aneka provides the essential platform. Aneka proves to be an extensible, flexible and to a great extent a market oriented cloud deployment and cloud application development solution using the .NET framework. Another important feature of aneka is that it equips the user with a SDK which can support multiple instances of programming model. It is essentially regarded as the market oriented cloud development as well as management framework. This title is owing to the fact that aneka allows user to create, schedule and monitor results using parameters like accounting, pricing and QoS or SLA (Service Level Agreement) for services on leased environments. Some of the lucrative feature of aneka are enlisted below :-

a) Rapid development and deployment of tools and framework

b) SLA/QoS based provisioning ability

c) Ability to run multiple programming instances and applications environment

d) In order to obtain application result quickly it can gain multiple physical or virtual machines[3]

D. Haizea (Latest Version : haizea 1.3.tar.gz)

In simple terms, haizea can be understood as a resource manager. Precisely, haizea is software which is employed to maintain a set of resources. These set of resources (often termed clusters) are managed by haizea and allocated to user as per requirement. It is obvious that an end user often asks for more than one resource at a time, in such scenario’s haizea comes to play and allocates the user a cluster or a part of it pertaining to the needs and the priority. It also termed as lease manager as mostly the resources are allocated to the user for a fixed time instant and for a fixed value of money, as per the contract of agreement. It also considered an appropriate replacement for OPENNEBULA. Some of the noticeable features of haizea are discussed below:-

a) Allows requests for one and those with more no. of VM's to run in parallel

b) Provides a token for advance reservation of leases

c) Leases marked as immediate are allowed to run instantly owing to priority issues without arising conflicts

d) Maintains most users as cluster; facilitates fats allocation.

XI. RESEARCH CHALLENGES AND FUTURE SCOPE

In this rapidly evolving computing world, resources are of utter importance. Therefore it becomes highly essential and mandatory to effectively manage these resources and a major part of it is dominated by effective scheduling of jobs on the resources. Thus, we need to find the most effective algorithm to schedule the jobs for the resources. As discussed earlier, there are several algorithms which have been employed for effective scheduling of resources. However, none of them is perfect. Each one suffers from one or more of their little flaws. These algorithms were designed to get the optimal or atleast nearly optimal solution of the problem. However, the current scenario demands of much more accurate algorithm for resource scheduling and resource allocation. The biggest research challenge is to discover such an algorithm.

There are various parameters which need to be considered, evaluated and modified in order to attain the goal, to develop a perfect job scheduling optimization algorithm. Some of the highly discussed research concerns are enlisted below:-

A. Un-interrupted connection throughout the scheduling

The cloud services are provided to the end user extend over large physical areas. Some of them are covered by wired links while others by wireless communication. In such scenarios, while we develop an efficient algorithm we also need to consider the challenges faced by wireless communication such as fading, interference, attenuation, noise etc. While mobile agents are communicating to design an efficient strategy for resource scheduling we should be prepared that none these problems hinder the process of scheduling. The scheduling algorithm should run while keeping a backtrack of the progress so far covered. So that in case of any hindrance the scheduling can resume from most effective point [6].

B. Overhead of scheduling

Another most important thing to be considered is the main goal of cloud computing. The main goal is to "provide optimal services that too at a minimal expense to the users". In order to achieve this it must be ensured that the resource scheduling algorithm undertaken not only benefits the cloud provider but also proceed in direction to provide optimal economic benefits to the end users also[16].

C. Robust and flexible

The resource scheduling technique must be robust and flexible at the same time. In case of any failure of VM's the jobs submitted by user must be re-scheduled on other VM's without disturbing the execution of other jobs. The scheduling algorithm should be such that it provides complete flexibility to the user. The user can anytime withdraw his job from the scheduling, however when he again submits task it must be re-initiated from the same point, where it was left [1].

Scheduling of jobs is a classical problem. All the possible options are considered while the others are explored. Scheduling meta-jobs (independent jobs) has been considered. Dynamic Acyclic Graphs, commonly referred as DAG's have been used to represent application tasks. [13]Dynamic approach in terms of techniques like [14] hybrid mapper and algorithms like [14] generational algorithms are being implemented. All the effort and research is directed with one notion, to achieve an algorithm which can provide optimal usage of the resources by effectively scheduling and maintaining them.

XII. CONCLUSIONS

In this paper, we have understood the role of efficient resource scheduling in order to utilise the resources in the best possible way. We have discussed various scheduling algorithms and how they work in cloud computing environment to attain the desired results. These algorithms were efficiently used in cloud and they have been immensely modified over the period of time to suit the needs of present cloud environment. Comparative analysis of these algorithms revealed their competitive features. There has been a lot work done to get the most effective resource scheduling algorithm and there is a lot more that still needs to be explored. Time will reveal the most desired algorithm which would be perfectly suited to achieve the goal.

REFERENCES

[1] A.H. Alhusaini and V.K. Prasanna, C.S. Raghavendra, “A Unified Resource Scheduling Framework for Heterogeneous Computing Environments”, HCW '99 Proceedings of the Eighth Heterogeneous Computing Workshop, 1999, pp 156 – 165.

[2] C.D. Chapman, M.J. Jakiela, “Genetic algorithm-based structural topology design with compliance and topology simplification considerations”. M.I.T, Journal Of Mechanical Design: vol. 118, n1: pp. 89-98 (24 ref.) June 01, 1994. doi:10.1115/1.2826862

[3] Cloud Book [2012] “Cloud Federation”, [Online]. Available: http://www.cloudbook.net/directories/research project.php?id=100004

[4] Cloudsim [2013] “Infrastructures, Services and features” [Online]. Available: *http://code.google.com/p/cloudsim/*

[5] Eberhart R.C., Shi Y., Kennedy J., “Swarm Intelligence”. 2001

[6] H. Fattah and C. Leung, “An Overview of Scheduling Algorithms in Wireless Multimedia Networks”, IEEE Wireless Communications, October 2002, 9(4), pp: 76 - 83

[7] H. Mühlenbein and D. Schlierkamp-Voosen, “Evolutionary Computation”. Journal of Evolutionary Computation, Spring 1993 by the Massachusetts Institute of Technogy: , Vol. 1, No. 1, December 10, 2007. doi>10.1162/evco.1993.1.1.25

[8] H.Zhong, K. Tao, X. Zhang, “An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems”, The Fifth Annual ChinaGrid Conference, July 2010. pp. 124 – 129. doi>10.1109/ChinaGrid.2010.37

[9] Ian Kenny, “Dynamic, Hierarchical Particle Swarm Optimization”, Technical report, 2008-2009.

[10] J.Alcaraz and C.Maroto “A Robust Genetic Algorithm for Resource Allocation in Project Scheduling”. Annals of Operations Research 102: pp 83-109, February 2001, Volume 102, Issue 1-4

[11] K. Li, G. Xu, G. Zhao, Y. Dong, D. Wang, “Cloud Task scheduling based on Load Balancing Ant Colony Optimization”, Sixth Annual ChinaGrid Conference, 22-23 Aug. 2011, pp 3-9. doi: 10.1109/ChinaGrid.2011.17

[12] K.Nishant, P. Sharma, V. Krishna, C. Gupta and K.P. Singh, Nitin and R.Rastogi, “Load Balancing of Nodes in Cloud Using Ant Colony Optimization”, 14th International Conference on Modelling and Simulation, March 2012, pp.3-8

[13] M. Iverson and F. Ozguner, “Dynamic, competitive scheduling of multiple DAGs in a distributed heterogeneous environment,” *7th Heterogeneous Computing Workshop (HCW’98)*, , March 1998**,** pp. 70-78. doi: 10.1109/HCW.1998.666540

[14] M. Maheswaran and H. J. Siegel, “A Dynamic matching and scheduling algorithm for heterogeneous computing systems,” *7th Heterogeneous Computing Workshop (HCW’98),* March 1998, pp. 57-69.

[15] M.B. Wall, “A Genetic Algorithm for Resource-Constrained Scheduling” – Ph.D Thesis, M.I.T., June 1996

[16] Q. Zhang, L. Cheng, R.Boutaba, “Cloud computing: state-of-the-art and research challenges”, Springer, May 2010, Volume 1, Issue 1, pp 7-18

[17] R. Mishra and A. Jaiswal, “Ant colony Optimization: A Solution of Load balancing in Cloud”, International Journal of Web & Semantic Technology(IJWesT), April 2012, Vol.3, No.2, pp. 33 - 50

[18] S. Banerjee, I. Mukherjee, and P.K. Mahanti, “Cloud Computing Initiative using Modified Ant Colony Framework”, World Academy of Science, Engineering and Technology (56) 2009, pp 221 – 224

[19] S. Hartmann, “A competitive genetic algorithm for resource-constrained project scheduling”, Naval Research Logistics, 7 DEC 1998, Volume 45, Issue 7 pp 733–750. doi: 10.1002/(SICI)1520-6750(199810)45:7<733::AID-NAV5>3.0.CO;2-C

[20] S. Pandey, L. Wu, S.M.Guru, R. Buyya, “A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments”, 24th IEEE International Conference on Advanced Information Networking and Applications, 20-23 April 2012, pp 400 – 407. doi: 10.1109/AINA.2010.31

[21] S. Tayal, “Task Scheduling Optimization for the Cloud Computing System”, (IJAEST) International Journal of Advanced Engineering Sciences and Technologies 5.2 (2011), pp. 111-115

[22] S.S. Leu, C.H. Yang, J.C.Huang, “Resource leveling in construction by genetic algorithm-based optimization and its decision support system application”, Automation in Construction 10-2000, Volume 10, Issue 1,pp 27–41. doi: http://dx.doi.org/10.1016/S0926-5805(99)00011-4

[23] Swarm Intelligence [2013] “Particle Swarm Optimization”. [Online]. Available: *http://www.swarmintelligence.org/*

[24] T. Viese , “‘Global optimization Algorithms – Theory and application”. Version: 2009

[25] Y. Shi, R. C. Eberhart, “Empirical Study of Particle Swarm Optimization”, Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999, Vol. (3)

[26] Z. Wu, Z. Ni, L. Gu, X. Liu, “A Revised Discrete Particle Swarm Optimization for Cloud Workflow scheduling”, International Conference on Computational Intelligence and Security, 11-14 Dec. 2010, pp 184 – 188.