Natural Computing: Performance Analysis of Evolutionary Algorithms for Task Scheduling in Cloud Environment

D. Sumathi[1], Dr. P. Poongodi[2]

[1] *Dept. of Computer Science and Engineering*
Jayam College of Engineering and Technology
Dharmapuri, Tamilnadu, India

[2]*Dept. of Electronics and Communication Engineering*
KIT-Kalaignarkarunanidhi Institute of Technology
Coimbatore, Tamilnadu, India

Abstract: *Cloud is ubiquitous, convenient virtual environment having pool of heterogeneous resources to provide services through Internet on pay-per-use, on-demand and at-scale basis. Cloud Scheduling algorithms used allocate ready tasks to available resources at affordable time to provide quality services to cloud users. Allocating resources in heterogeneous and geographically distributed environment is NP-hard. Many swarm based meta heuristic algorithms have been proposed by various researchers to solve these problems. In this paper, we analyse some Natural Computing Algorithms which inspired by the social behaviour of animals, birds and insects to solve complex mathematical problems and provides better solution.*

Keywords — *Cloud Computing, Scheduling Algorithm, Natural Computing.*

I. Introduction

Cloud computing aims to offer virtualized and distributed resources with elasticity to the end users. The enterprises utilize the virtualization technology to lease the computing resources which has been full realistic and flexible to the customers. Hence more number of users can access the large number of virtual resources. Cloud scheduling algorithms assigns the tasks to the corresponding virtual machines to provide better services in a huge distributed cloud environment. The traditional scheduling algorithms like first-come-first-service, round-robin, shortest-job-first and etc. are not suitable in cloud environment to assign volumes of tasks to the corresponding Virtual Machines (VM).

First Come First Served scheduling algorithm (FCFS) is a simple and fast in which the job in the row that become first is assisted. In Round Robin scheduling algorithm (RR), the procedures are remitted in a FIFO means with the restricted quantity of CPU time named as a time-slice or a quantum. Before the CPU-time expires, if process is not complete then CPU is pre-empted to the subsequent procedure to come in a row where the pre-empted course has been located at the ready list end. Min–Min algorithm a small task for execution initially where the large one intervals for lengthy time. Max – Min algorithm initially executes large tasks where minor task delays for long time [1].

Dynamic scheduling algorithms ensures higher performance than static algorithms but results are high overhead. Scheduling in cloud is NP-hard as the virtual machine parameters are dynamic in nature and resources also heterogeneous [1]. Several heuristics algorithms proposed to solve the NP-hard scheduling and resource allocation problem.

II. Cloud Scheduling Algorithms

Cloud scheduling is a set of policies which controls the work order. Scheduling algorithms of different kinds have realistic on several data assignments with various routing metrics to perform evaluation. Most of scheduling algorithms improves the Quality of services to execute jobs and to offer the assumed outcome on time and maintained efficacy and fairness in all jobs. [4]

There are three stages in cloud scheduling [3]

1) Resource discovering and filtering - DataCenter Broker discovers resources in a network system and collects status information about them.

2) Resource selection - Based on certain parameters of task and resource, target resource is selected. This is deciding stage.

3) Task submission - Task submitted to selected resource

Many new algorithms have been developed to overcome the NP-hard problem based on trial and error which produce optimum solution. Optimization is used to find the best solution or result under given circumstances and maximize or minimize the value of a function to act as a local or global optimum. Different types of problems are utilized like linear optimization, non-linear optimization, dynamic optimization etc. and all have different technique for solving the above said problem.

In this paper, we discussed various swarm-based algorithms like Ant-Colony Optimization, Particle Swarm Optimization, Honey Bee Optimization, Firefly algorithm, Cuckoo search algorithm, Bat Algorithm, Termites Algorithm and Wasp Algorithm. These algorithms are developed based on the social behavior of those animals and insects.

III. SWARM BASED ALGORITHMS

Swarm based algorithms are otherwise called as Swarm intelligence (SI), is the animals social behavior of decentralized, self-organized nature to identify food or to build home to live. The concept is employed in work on artificial intelligence. The expression was introduced by Gerardo Beni and Jing Wang in 1989, in the context of cellular robotic systems [10]. Methods, mechanisms as well as protocols which the domain adopts are inspired by the behavior of insects, birds, as well as fish, and their unique capability for solving complicated tasks in the form of swarms, though the same task would not be possible in the individual level. Individual ants, bees or birds have limited intelligence when they act alone, however through social interaction with one another as well as their environment, they are capable of accomplishing difficult tasks like discovering the shortest route to a food source, organizing the nest, synchronizing their movement, as well as travel as one coherent entity with higher speed.

A. Ant Colony Optimization (ACO) Algorithm

This algorithm simulates the scavenging ant colonies characteristics. The ants are using special kind of chemical named pheromone for their communication. Randomly they search food at first and if the way to food source is found, they consent path pheromone. All ants track the path through identify in ground pheromone so that the most of ants selects the minimum distance path as a large pheromones values collected on this path. It is advanced of its optimistic opinion appliance, inner parallelism and extensible [5]. The algorithm for ACO with a set of cloudlets and VMs to get a best solution for task allocation.

Step 1: Initialize: Set Current_iteration_t=1.
Set Current_optimal_solution=null.
Set Initial value τij(t)=c for each path between tasks and VMs.
Step 2: Place m ants on the starting VMs randomly.
Step 3: For k =1 to m do Place the starting VM of the k-th ant in tabuk .
Do ants_trip while all ants don't end their trips
Every ant chooses the VM for the next task.
Insert the selected VM to tabuk .
End Do
Step 4: For k=1 to m do Compute the length Lk of the tour described by the k-th ant.
Update the current_optimal_solution with the best-founded solution.
Step 5: For every edge (i, j), apply the local pheromone.
Step 6: Apply global pheromone update.
Step 7: Increment Current_iteration_t by one.
Step 8: If (Current_iteration_t < tmax) Empty all tabu lists.
Goto step 2 Else Print current_optimal_solution.
End If
Step 9: Return

The disadvantages are above head and the inactivity occurrence or penetrating to a particular range. Each individual has identified all resolution precisely that cannot be found further to the solution area to make this algorithm to meet local optimum solution.

B. Particle Swarm Optimization (PSO) Algorithm

The Particle Swarm Optimization algorithm [3] is a swarm-based intelligence algorithm predisposed through social activities of animals like birds' flock to discover a food source and a school of fish preserving from a predator. A particle is similar to a bird or fish flying in a search space and all the particles movement has been synchronized through a velocity of magnitude and direction. All particles positions are effected by their better problem space position. Its presentation is restrained based on its fitness function has a -particular problem. The particle numbers present in a problem space is the population in which the particles are primed arbitrarily and the fitness value is evaluated by fitness function to be optimized in all generation. All the particles have its best position named as pbest and the best position amid the entire group of particles called as gbest. The best fitness value stretched by a particle is the pbest where best particle in an entire population is the gbest [7]. The PSO scheduling algorithm is given below.

Step 1: Set particle dimensions equal to the size of ready tasks in $\{t_i\} \in T$
Step 2: Initialize particles position randomly from PC=1, ..., j and velocity v_i randomly.
Step 3: Calculate its fitness value
Step 4: Compare fitness value with previous best pbest
Step 5: If the fitness value is better than the previous best pbest, set the current fitness value as the new pbest.
Step 6: After step 3 and 4 for all particles, select the best particle as gbest.
Step 7: Update velocity and positions.
Step 8: If the stopping criteria or maximum iteration is not satisfied, repeat from step 3.
Step 9: End.

C. Honey Bee Optimization (HBO) Algorithm

A colony of honey bees can extend itself over long distances and in multiple directions simultaneously to harvest nectar or pollen from multiple food sources (flower patches). A small fraction of the colony constantly searches the environment looking for new flower patches. These scout bees move randomly in the area surrounding the hive, evaluating the profitability (net energy yield) of the food sources encountered. When they return to the hive, the scouts deposit the food harvested. Those individuals that found a highly profitable food source go to an area in the hive called the "dance floor", and perform a ritual known as the waggle dance. Through the waggle dance a scout bee communicates the location of its discovery to idle onlookers, which join in the exploitation of the flower patch. Since the length of the dance is proportional to the scout's rating of the food source, more foragers get recruited to harvest the best rated flower patches. After dancing, the scout returns to the food source it discovered to collect more food. As long as they are evaluated as profitable, rich food sources will be advertised by the scouts when they return to the hive. Recruited foragers may waggle dance as well, increasing the recruitment for highly rewarding flower patches. [7] The algorithm is as below

Step 1: Start the AWS EC2 and login in and after that pair the key. If the key is correct then create instances and
if key is wrong then generate new key and again login
Step 2: After instances are created assign volume to them and then create load balancer and set number of cloudlets
Step 3: Assign initial load on VM=null and then send first request by cloudlet
Step 4: Generate scout for each VM
Step 5: Send the request. It will check whether the
current load is <= to threshold value or not
Step 6: If it is less than or equal to threshold value then it will assign task to current VM or it send scout
to find new VM randomly
Step 7: After that it will check whether all scout have searched all VM
Step 8: If yes then it will assign task to VM which has min. load or If no then again it will check
whether current load is <= threshold or not
Step 9: After that it will calculate cost

D. Cuckoo Search (CS) Algorithm

The Cuckoo Search algorithm [3] is an optimization algorithm established by the brood parasitism of few cuckoo kinds by resting their eggs in its nests of another host birds. Direct conflict is involved by some host birds with the interfering cuckoos. If a host bird determines that the egg is not their own, then it throw away the unfamiliar eggs or recklessness its nest by building new one. The character of cuckoo is described as good than other animals which have extensive use in computing Intelligence Systems. An initial set of nests represents the solution and these solutions are updated over multiple generations. The process updating of an individual solution is the choosing of random nest. By random walking the new solutions are generated. The new solution is replaced by different solution chosen as random if its fitness value is better than the original and ranking of fitness value is done in which the worst rank is replaced with random solutions. This combination is allowed to search locally and globally at the same time for the optimal solution.

Step 1: Initialization
Objective function f(x), x= {x1,x2,…, xd}
Generate an initial population of 'n' host nests
xi, i = 1, 2, …, n
Step 2: Updating loop
While (Stop Criterion)
Select a cuckoo bird (i) randomly using levy Flights
Find its Fitness function Fi, Select a nest (j)
randomly among (n)
If (Fitness i > Fitness j) then Replace j by i
End If
A Probability (Pa) of worst nest is removed.
Build the new nest
Record the best solutions
Sort these solutions and find current best
End While
Step 3: If solution is optimum then is result.
Step 4: Else Pass the Best solution to next iteration then repeat Step 2:
Step 5: End Loop

E. Bat Algorithm (BA)

The Bat Algorithm (BA), [8] is inspired by the research on the social behaviour of bats. The BA is based on the echolocation behaviour of bats. Microbats use a type of sonar (echolocation) to detect prey, avoid obstacles, and locate their roosting crevices in the dark. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. Their pulses vary in properties and can be correlated with their hunting strategies, depending on the species. The structure of the pseudo code of the Bat Algorithm is as follows.

Step 1: Initialize Bat population its position and velocities
Step 2: Echolocation parameters and their initializations
Step 3: Evaluating micro-bats in initial position, the calculated solutions are stored in resource log
Step 4: Sorting the current population in descending order
Step 5: Generate candidate micro-bats.
Step 6: Random flying by adjusting new position by adjusting new position, velocity and direction.
Step 7: Randomly select a micro-bat from population
Step 8: Evaluate micro-bat by fitness function

Step 9: Echolocation parameters update for better next position
Step 10: Select the microbat for next iteration
Step 11: Repeat step 4 to step 10 until reach terminate criterion.

F. Firefly Algorithm (FA)

Firefly Algorithm (FA) [9] is a metaheuristics scheduling algorithm based on the flashing patterns and intelligent behaviour of fireflies. FA uses the following three idealized rules

- Firefly flashing light to attract their mate.
- The attractiveness is based on the brightness of light, the less-brighter firefly moves towards to more-brighter one.
- The brightness is based on the distance and determined by the objective function.

The FA is suitable to solve the NP-hard problem of cloud scheduling algorithm

Step 1: objective function f(x), x = (x1, x2, …, xn)
Step 2: Generate initial population of firefly xi, i = 1,2,…,n)
Step 3: light intensity li determined by f(xi)
Step 4: light absorption coefficient γ
Step 5: while t < G_{max} for all fireflys evaluate flashing light intensity
Step 6: move the firefly xi towards j in d-dimension
Attractiveness varies with distance γ_{ij} via $e^{-\gamma rij}$
Step 7: new solution and update light intensity.
Step 8: rank the fireflys and find current best.
Step 9: repeat step 5 to step 9 until stop criteria.

G. Termite Algorithm (TA)

Termites are small and simple beings and incapable to do complex tasks individually. The termite colony on the other hand seen an intelligent entity for its great level of self-organization and performing the complex tasks. Termites are communicated with each other by a kind of pheromone which emits from its body. Termites are capable of identifying if another termite is member of its own colony or different by the smell of its body. The pheromone once emitted then it starts evaporate. Individual termites leave a trail of such scent, which stimulates other termites to follow that trail, dropping pheromones while doing so. This process will continue until a trail from the termite colony to the food source is established. One of the characteristics of the pheromone trail is that it is highly optimized, tending toward the shortest highway between the food source and the termites' nest (hill)

IV. Performance Analysis of Swarm based Algorithms

The Ant Colony Optimization (ACO) [9] algorithm performs effectively in dynamic scheduling and also used to solve NP-hard issue in cloud scheduling. ACO produces a guaranteed solution in large number of tasks and resources. The disadvantage is the time complexity of ACO is more when compared to other algorithms.

The Particle Swarm Optimization (PSO) is a heuristic protocol which has been employed in the solution of scheduling problems as well as other NP-hard issues. It is comparatively simple to implement when compared with Ant Colony Optimization (ACO) or Genetic Algorithm (GA). PSO has several benefits like being simple to understand, simple operations, as well as rapid searching. The disadvantage of PSO is in the resolution of a huge complicated issue, it is forced into local optima.

Cuckoo Search (CS) algorithm has benefits like fewer control variables as well as higher efficacy. However, it also has some drawbacks, like slower convergence speeds and lower accuracy. In CS, higher arbitrariness of Lévy flight means the search procedure rapidly jumps from one region to the next. Hence, the global search capacity of the protocol is extremely strong. But, because of the higher arbitrariness of Lévy flight, the protocol begins a blind search procedure; convergence speed becomes slower, while the searching efficacy is also considerably decreased closer to the optimum solution.

The Honey Bee Optimization (HBO) algorithms use standard evolutionary or random explorative search to locate promising locations. Then the algorithms utilize the exploitative search on the most promising locations to find the global optimum. The disadvantage of HBO is the random initialization. It has several control parameters and need to be turned often.

The Bat Algorithm (BA) is accurate and very efficient algorithm. The nature of automatic zooming, effective parameter control and the frequency turning and echolocation are great things to solve wide range of problems with quick time in promising optimal solution. It is suitable for complex problem. The disadvantage of this algorithm is it converge very quickly at early stage, and convergence rate will slow down. In large scale applications the accuracy is limited, and no mathematical analysis to link the parameters with convergence rate.

The Firefly Algorithm (FA) is having high convergence rate and robust. Also finds optimum solution in less population. The good information-sharing mechanism which can promote the algorithm to converge faster under certain conditions and the lower probability of entrapment into local modes. The disadvantage of FA is, in need of proper setting of dependent parameters and having huge number of iterations.

V. Conclusion

The evolutionary algorithms in Natural Computing algorithms are very effective and useful to solve very complex mathematical problems. The swarm-based optimization algorithms are now focused to solve NP-hard issues in cloud task scheduling. In this paper we analyzed various natural computing swarm-based algorithms like ACO, PSO, CS, HBO, BA, FA and TA algorithms based on allocation of ready tasks to available resources in complex cloud scheduling.

References

[1] Salot, P 2013, 'A Survey of Various Scheduling Algorithm in Cloud Computing Environment'. International Journal of research and engineering Technology (IJRET), ISSN, pp. 2319-1163.

[2] Mohialdeen, IA 2013, 'Comparative Study of Scheduling Algorithms in Cloud Computing Environment'. Journal of Computer Science, vol. 9, no. 2, p. 252

[3] D. Sumathi and P. Poongodi, Scheduling Based on Hybrid Particle Swarm Optimization with Cuckoo Search Algorithm in Cloud Environment, IIOAB JOURNAL 7 (9), 358-366, 2016

[4] Gupta, H., Singh, D., & Gupta, B. K. Scheduling Techniques in Cloud Computing: A Systematic Review

[5] Tawfeek, MA, El-Sisi, A, Keshk, AE & Torkey, FA 2013, 'Cloud Task Scheduling Based on Ant Colony Optimization'. In Computer Engineering & Systems (ICCES), 8th International Conference, IEEE, pp. 64-69

[6] D. Sumathi and P. Poongodi, Secure Medical Information Processing in Cloud: Trust with Swarm Based Scheduling, Journal of Medical Imaging and Health Informatics Vol. 6, 1636–1640, 2016

[7] Pandey, S, Wu, L, Guru, SM & Buyya, R 2010, 'A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments'. In Advanced information networking and applications (AINA), 24th IEEE international conference, IEEE, pp. 400-407

[8] Xin-She Yang, Bat Algorithm: Literature Review and Appliation', International Journal of Bio inspired computation, vol. 5, No. 3, 2013

[9] Xin-She Yang, 'Natural Inspired Meta Heuristics Algorithms', Luniver Press, ISBN:1905986106 9781905986101, 2008

[10] Beni, G., Wang, J. Swarm Intelligence in Cellular Robotic Systems, Proceed. NATO Advanced Workshop on Robots and Biological Systems, Tuscany, Italy, June 26–30 (1989)