INTERNATIONAL JOURNAL FOR TRENDS IN ENGINEERING & TECHNOLOGY,
VOLUME 3 ISSUE 2 – FEBRUARY 2015 – ISSN: 2349 – 9303

Comparative Study of Ant Colony Optimization And Gang Scheduling

Sridhar.S[1]
[1]B.S Abdur Rahman University, Computer Science,
007sridol@gmail.com

Sharon priya.S[2]
[2]B.S Abdur Rahman University, Computer Science,
Sharonpriya2004@gmail.com

Abstract— Ant Colony Optimization (ACO) is a well known and rapidly evolving meta-heuristic technique. All optimization problems have already taken advantage of the ACO technique while countless others are on their way. Ant Colony Optimization (ACO) has been used as an effective algorithm in solving the scheduling problem in grid computing. Whereas gang scheduling is a scheduling algorithm that is used to schedule the parallel systems and schedules related threads or processes to run simultaneously on different processors. The threads that are scheduled are belonging to the same process, but they from different processes in some cases, for example when the processes have a producer-consumer relationship, when all processes come from the same MPI program.

Index Terms**—** ANT-GA, Grid computational, Grid Computing, job completion, job scheduling, meta-heuristic.

1. INTRODUCTION

Grid computing overcomes the limitation that subsists in conventional shared computing mood, and becomes a principal trend in distributed computing system. Grid core service is the Centrum of entire grid computing which takes charge of entire grid system in order to ensure grid system work effectively, it is an important part of grid computing, and task scheduling technology is a part of grid core service technology. When grid resources are required by lots of tasks, the system can optimize the resources only by scheduling the tasks reasonably. In this paper we had compared Ant Colony Optimization and Gang Scheduling. Consequently, this paper attempts the measurement of Average Waiting Time and Average Response Time of both Ant Colony Optimization and Gang Scheduling. In accordance with the information gathered during the course of our research, the performance of the Gang Scheduling better than the Ant Colony Optimization.

I. ANT COLONY OPTIMIZATION

Entomologists have discovered that the real power of ants resides in their colony brain. ACO is inspired by a colony of ants that work together to find the shortest path between their nest and food source. ACO is based on the concept of stigmergy – indirect coordination between the agents or may be actions. The principle is that the trace left in the environment by an action stimulates the performance of a next action, by the same or a different agent. An example of stigmergy is the communication of ants during the foraging process: ants indirectly communicate with each other by depositing pheromone trails on the ground and thereby influence the decision process of other ants.

II. BASIC CONCEPT OF ANT COLONY

1) The ants walk to and far away from the nest to get the food, while moving they depositing a substance called “pheromone” on their path.

2) So that next ant that follow the first one is able to smell the pheromone and it influences the choice of their paths as the ants are able to follow stronger pheromone concentrations.

3) The pheromone that are deposited on the ground forms the pheromone trail. This allows all ants to find the sources of food that have previously been identified by their colony ants.

4) The pheromone keeps on evaporating it stays only for a short time period.

5) So that the ant can able to select the shorter path and it would also be the first one to return to the nest. This function is due to a very high probability of the ant choosing the same shorter path on its return.

6) Therefore, finally after some time, all the colony ants converge to follow the shortest path that has been choosed.

Below mentioned diagram is the procedure how the ant tour will start and execute. This is the basic concept of the ant to find its food source. Simultaneously the ant start from the nest and randomly chooses there root. While moving it deposited the pheromone so that next ant will smell that and follow the same root. The pheromone that is deposited by the ant will stay for a little time period so that the next ant also deposited the pheromone to follow the next ant. Likewise all ants will deposit the pheromone to make a local pheromone update. When the pheromone updated is completed all ants finds the best route. And then all ant will follow the route, they finally do the Global update. When there convergence criteria is met the tour get end. And all ant moves in the shortest path.

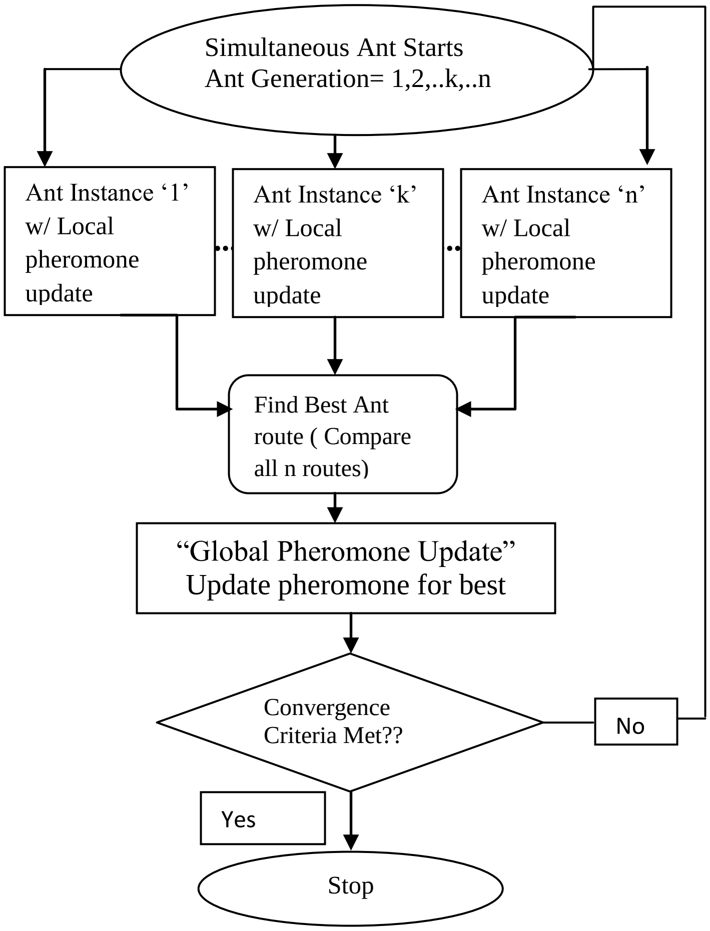

Simultaneous Ant Starts
Ant Generation= 1,2,..k,..n
Ant Instance '1' w/ Local pheromone update
Ant Instance 'k' w/ Local pheromone update
Ant Instance 'n' w/ Local pheromone update
Find Best Ant route (Compare all n routes)
"Global Pheromone Update" Update pheromone for best
Convergence Criteria Met??
No
Yes
Stop

Fig. 1. Ant colony optimization.

III. GANG SCHEDULING

Gang scheduling is used so that if two or more threads or processes communicate with their own processes or threads, and they will all be ready to communicate at the same time. If the processes or thread are not gang-scheduled, then at least one processes or one thread could wait to send or receive a message to another while the threads are sleeping, and vice-versa. When processors are over-flows and gang scheduling is not used within a group of processes or threads which communicate with their own processes or threads, it can lead to situations where each communication event suffers the overhead of a context switch.

Technically, gang scheduling is based on a data structure like in matrix (ousterhost matrix) form. In this type of matrix each row represents a time, and each column represent a processor. The threads or processes of one job are packed into a single row of the matrix. During the time of execution, the coordinated context switching is performed across all nodes to switch from the processes in one row to those in the next row.

Gang scheduling is stricter than coscheduling. It requires all threads of the same process to run in side by side, while coscheduling allows for fragments(to split a job into smaller parts), which are sets of threads that do not run concurrently with the rest of the gang.

Gang scheduling was implemented and used for several parallel machines, most notably the Connection Machine CM-5. [12]

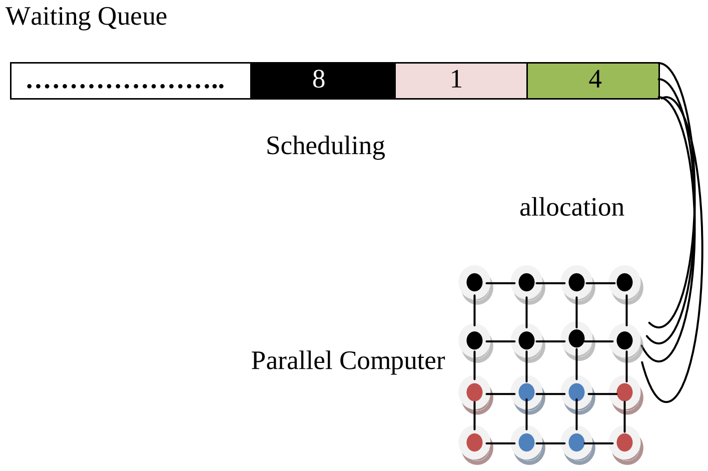


Fig. 2. Gang Scheduling frame work.

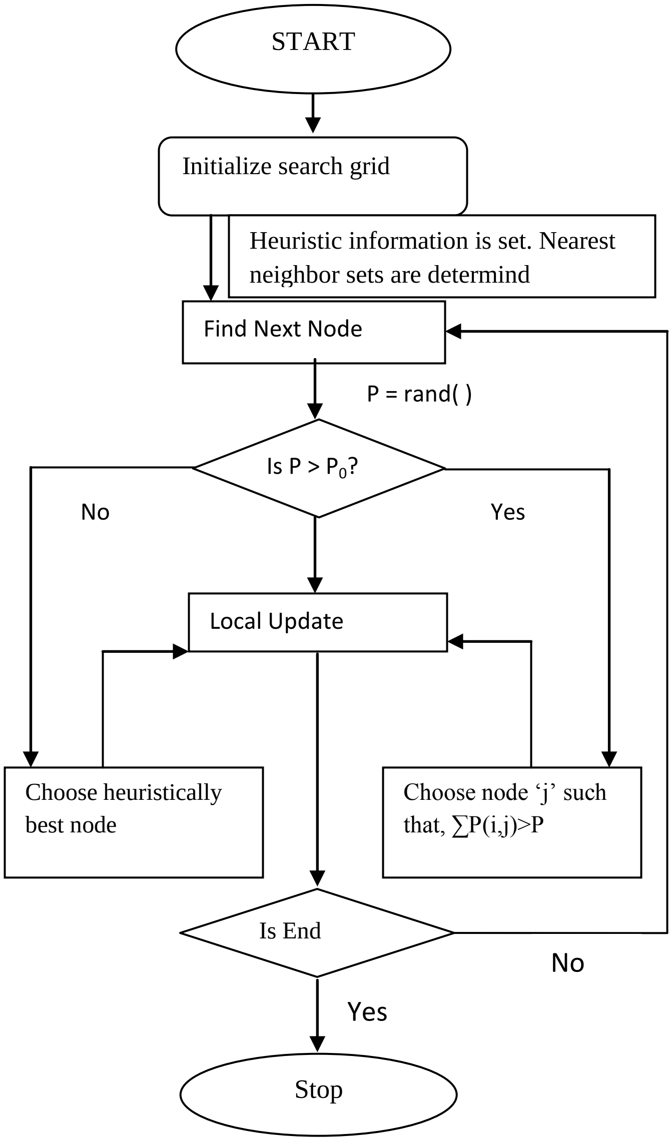

START
Initialize search grid
Heuristic information is set. Nearest neighbor sets are determind
Find Next Node
P = rand( )
Is P > P_0?
No
Yes
Local Update
Choose heuristically best node
Choose node 'j' such that, ∑P(i,j)>P
Is End
No
Yes
Stop

Fig. 3. Working Procedure Of GANG SCHEDULAR

IV. PERFORMANCE COMPARISON OF GANG SCHEDULING AND ANT COLON OPTIMIZATION:

Ant Colony Optimization use a hybrid evaluation measure that is able to estimate the overall performance of subset as well as the local importance of resources. A scheduling algorithm is used to estimate the performance of subset. In the first iteration, each ant will randomly choose a resource subset of m resources. Only the best subsets, be used to update the pheromone trail and influence the resource subset of the next Units iteration. In the second and following iterations, each ant will start with m-p resources that are randomly chosen from the previously selected best subsets where is an integer that ranges between 1 and m-1 . In this way, the resources that constitute the best subsets will have more chance to be present in the subsets of the next iteration. For experimental task some standard parameter are used such as resource characteristics and job characteristic.

Gang scheduling combines time-sharing with space-sharing to ensure a short response time for interactive tasks and high overall system throughput. It has been widely studied in different areas including the Grid and Cloud. Gang scheduling tries to assign the task belonging to one job to different Cloud nodes. During the tasks assignment, there are three targets as follows: (1) to keep the Cloud in higher resource utilization, (2) to keep the jobs in a low average waiting time and executing time, and, (3) to keep the system in fairness between jobs.

Gang scheduling is an efficient time–space sharing scheduling algorithm that is used to effectively schedule frequently communicating parallel tasks in distributed systems, multi-core clusters, parallel systems, Grid, and Cloud computing. In this way, eliminate the risk of waiting for a task that is currently not running on a resource node. The number of tasks in a Gang cannot exceed the number of available resource nodes because there must be a one-to-one mapping between tasks and processors.

Whereas in ACO there is no one-to-one mapping between tasks and processors. It consists of time and space sharing also. Gang scheduling seek a way to maximize the performance of the system by avoiding unnecessary delays. The system which uses the Gang Scheduling algorithm every task of a given job allocated to different resource node, and it will begin and finish its execution at the same time. By allocating like this the system can avoid the cases where a task is blocked while waiting for the input from another unexecuted task.

Gang scheduling enables processes in the same job to run at the same time. This leads to better performance for compute-bound communicating processes. So that, I/O-bound processes cause the CPUs to be idle for some long time period, while there are other processes that are able to run. At the same time, the I/O-bound processes makes the disks to be busy, while compute-bound processes leave them idle.

Whereas the process taken by ACO will consider the pheromone value which depends on the time taken by each resource to perform the process of the jobs. It does not consider the capacity of resources such that their bandwidth to perform, their processor speed and their load.

The main drawback of Ant Colony is that jobs are not scheduled efficiently and therefore load among the resources are not balanced. This problem can be fixed by increasing the number of ants that can explore the entire grid system to find resources with the lightest load. An ACO algorithm for load balancing in distributed systems is done through the use of multiple ant colonies.

Multiple ant colonies have been adopted such that each node will send a colored colony throughout the network. Colored ant colonies are used to prevent ants of the same nest from following the same route and also enforcing them to be distributed all over the nodes in the system and each ant acts like a mobile agent which carries newly updated load balancing information to the next nodes.

The pheromone update function in this research is performed by adding, punishment coefficient, encouragement and load balancing factor. The initial pheromone value of one resource is based on its status where a job is assigned to the resource with the maximum pheromone value. The strength of pheromone of each resource will be updated after completion of the job. The encouragement, punishment and local balancing factor coefficient are defined by users and are used to update pheromone values of resources.[11]

If a resource completed a job in success, more pheromone get added by the encouragement coefficient in order to be selected for the next job to perform execution. If a resource failed to complete a job, then it will be punished by adding low number of pheromone. The load of each resource is taken into account and the balancing factor is also applied to change the pheromone value of each resource.

Whereas in Gang Scheduling job is assigned to the every node, each resources must have only one job to be allocated. If more job is allocated the resources will terminate and not used. For allocating the job to the resources one must follow the above three targets assigned by the Gang Scheduling algorithm. So that there is no overlap of job. After the allocation of the job is over and it gives a best performance.

If the resource completed a job successfully, more job is added to get the execution in less waiting time and less response time. So that Gang scheduling outperforms well when compared to ACO.

The local and global pheromone update techniques are used to balance the load of the system. The (local) pheromone update function updates the status of the selected resource after a job has been assigned and the job scheduler depends on the latest information of the selected resource for the next job to be submitted. The (Global)pheromone update function updates the status of each resource for all jobs after all jobs completed. By using these two update techniques, the job scheduler will get the latest information of all resources for the next job submission.

The term "gang scheduling" refers to all of a program's threads of execution being grouped into a gang and side by side scheduled on dissimilar processors. Furthermore, time-slicing is supported through the concurrent preemption and later rescheduling of the gang. These threads of execution are not necessarily POSIX threads, but components of a program which can execute at a same time. The threads may span multiple computers and/or UNIX processes. Communications between threads may be performed through sharing the memory, by passing the message, and/or other means.

According to the paper Gang Scheduling perform well when compared to Ant Colony Optimization. So as per my evaluation

Gang Scheduling is the best scheduling algorithm when compared to Ant Colony Optimization. A gang scheduling algorithm may be composed of four parts: a packing scheme, a re-packing scheme, a queue policy and a multiprogramming level. So that it is so simple and easy scheduling algorithm when compared to Ant Colony Optimization. In this case, gang scheduling has two parts namely space sharing and time sharing, space is related with the parallelism degree and time with the: number of instructions, granularity and the other factors. Combining space (parallelism degree) and time (execution time), we can cover the majority of possible workloads.

Granularity: Low – 1 million instructions, High – 10 million instructions.
Number of Instructions: Low – 100 million instructions, High – 1 billion instructions.
Parallelism Degree: Low – uniform distribution (1,4), High – uniform distribution (5,16).
Parallel Algorithm Model: Process Farm (Master Slave).
Message Size: 16 Kbytes.[14]

Table 1. Advantages and Disadvantages of Ant Colony Optimization

Ant colony algorithm:	
Advantage	Disadvantage
1.Inherent parallelism 2.Positive Feedback accounts for rapid discovery of good solutions 3.Effcient for Traveling Salesman Problem and similar problems	1.Theoretical analysis is diffcult 2.Sequences of random decisions (not independent) 3.Probability distribution changes by iteration 4.Research is experimental rather than theoretical 5. Time to convergence uncertain (but convergence is gauranteed!)

Likewise the Gang Scheduling Algorithm advantages are as follows:

Table 2. Advantages and Disadvantages of Gang Scheduling

Gang Scheduling	
Advantages	Disadvantages
1. Gang scheduling are similar to those of time-sharing in uniprocessor systems.	Gang scheduling still has the problem of the fragmentation , and the excessive number of time slots.
2. High system utilization can be sustained under a wide range of workloads.	2. Context switch time, a CPU is unavailable while a program's image is being transferred between disk and memory.

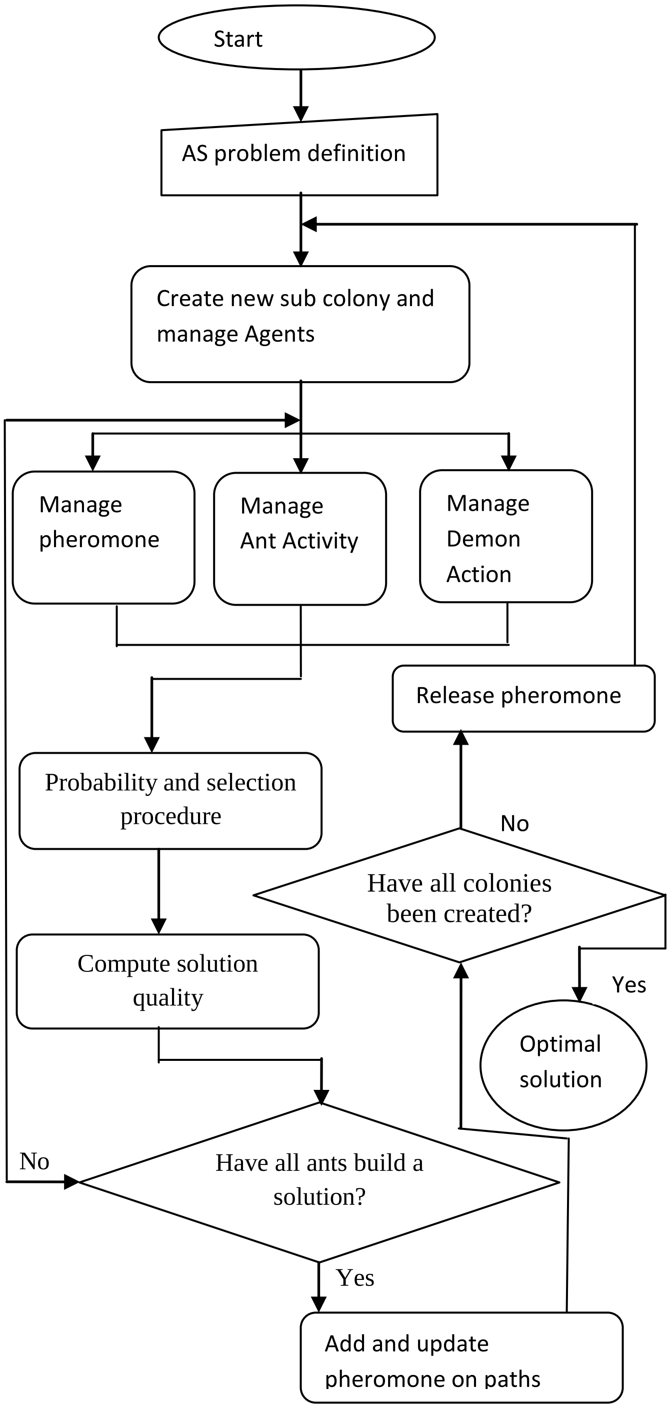
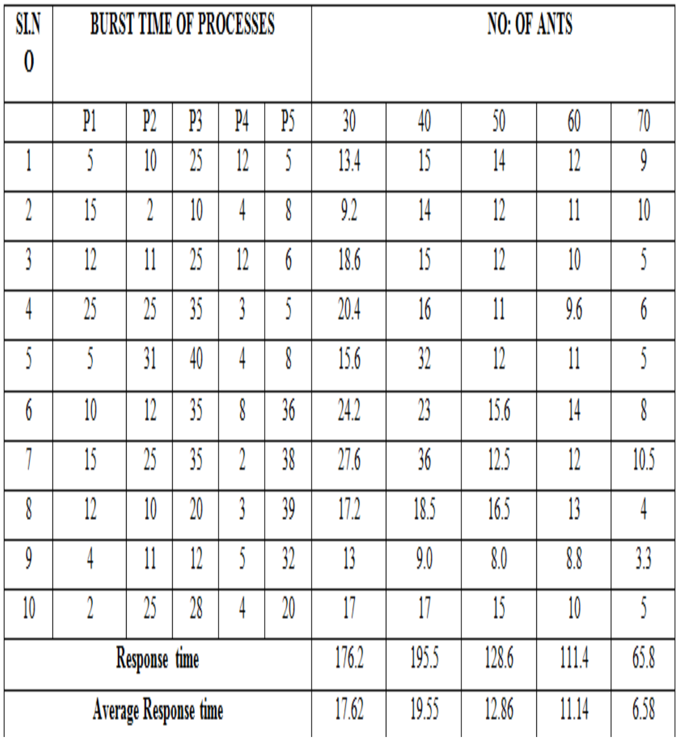

Start
AS problem definition
Create new sub colony and manage Agents
Manage pheromone
Manage Ant Activity
Manage Demon Action
Release pheromone
Probability and selection procedure
No
Have all colonies been created?
Yes
Compute solution quality
Optimal solution
No
Have all ants build a solution?
Yes
Add and update pheromone on paths

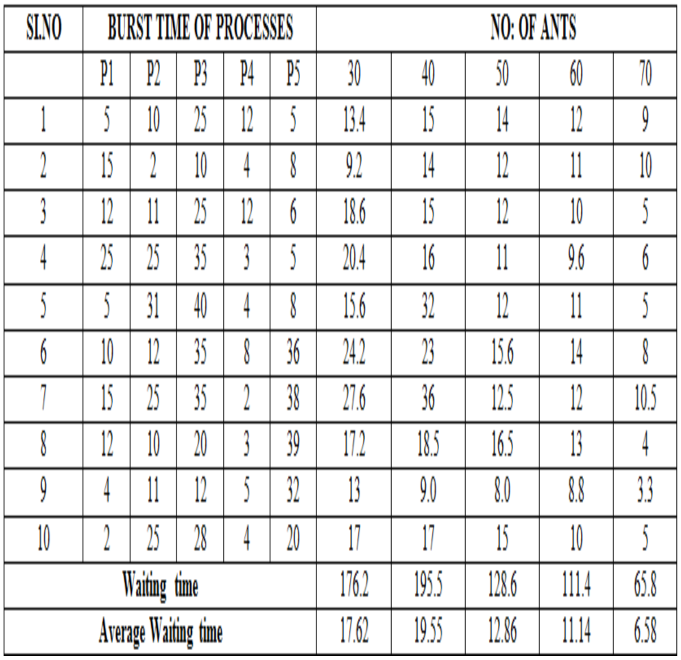
Fig. 4. Workflow diagram of Ant Colony Optimization

V. ANT COLONY OPTIMIZATION: IF NO OF ANTS INCREASES RESPONSE TIME IS MINIMIZED. [Table 3. Response Time Calculation of ACO]



SL.N 0	BURST TIME OF PROCESSES					NO: OF ANTS				
	P1	P2	P3	P4	P5	30	40	50	60	70
1	5	10	25	12	5	13.4	15	14	12	9
2	15	2	10	4	8	9.2	14	12	11	10
3	12	11	25	12	6	18.6	15	12	10	5
4	25	25	35	3	5	20.4	16	11	9.6	6
5	5	31	40	4	8	15.6	32	12	11	5
6	10	12	35	8	36	24.2	23	15.6	14	8
7	15	25	35	2	38	27.6	36	12.5	12	10.5
8	12	10	20	3	39	17.2	18.5	16.5	13	4
9	4	11	12	5	32	13	9.0	8.0	8.8	3.3
10	2	25	28	4	20	17	17	15	10	5
Response time						176.2	195.5	128.6	111.4	65.8
Average Response time						17.62	19.55	12.86	11.14	6.58

VI. ANT COLONY OPTIMIZATION: IF NO OF ANTS INCREASES WAITING TIME IS MINIMIZED. [Table 4. Waiting Time Calculation of ACO]



SLNO	BURST TIME OF PROCESSES					NO: OF ANTS				
	P1	P2	P3	P4	P5	30	40	50	60	70
1	5	10	25	12	5	13.4	15	14	12	9
2	15	2	10	4	8	9.2	14	12	11	10
3	12	11	25	12	6	18.6	15	12	10	5
4	25	25	35	3	5	20.4	16	11	9.6	6
5	5	31	40	4	8	15.6	32	12	11	5
6	10	12	35	8	36	24.2	23	15.6	14	8
7	15	25	35	2	38	27.6	36	12.5	12	10.5
8	12	10	20	3	39	17.2	18.5	16.5	13	4
9	4	11	12	5	32	13	9.0	8.0	8.8	3.3
10	2	25	28	4	20	17	17	15	10	5
Waiting time						176.2	195.5	128.6	111.4	65.8
Average Waiting time						17.62	19.55	12.86	11.14	6.58

VII. GANG SCHEDULING: (WAITING TIME)

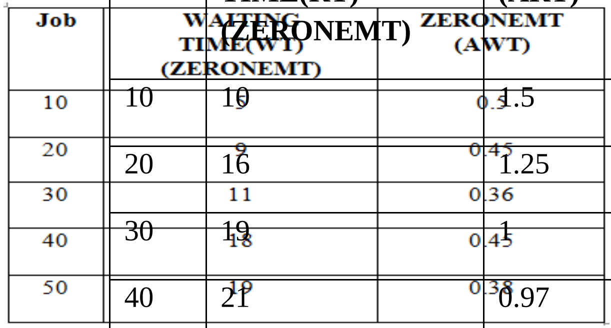
[TABLE 5. Waiting time Calculation of Gang]

Job	WAITING TIME(WT) (ZERONEMT)	ZERONEMT (AWT)
10	5	0.5
20	9	0.45
30	11	0.36
40	18	0.45
50	19	0.38

VIII. GANG SCHEDULING:(RESPONSE TIME)

[Table 5. Response Time Calculation of Gang]

Job	RESPONSE TIME(RT) (ZERONEMT)	ZERONEMT (ART)
10	10	1.5
20	16	1.25
30	19	1
40	21	0.97
50	31	1

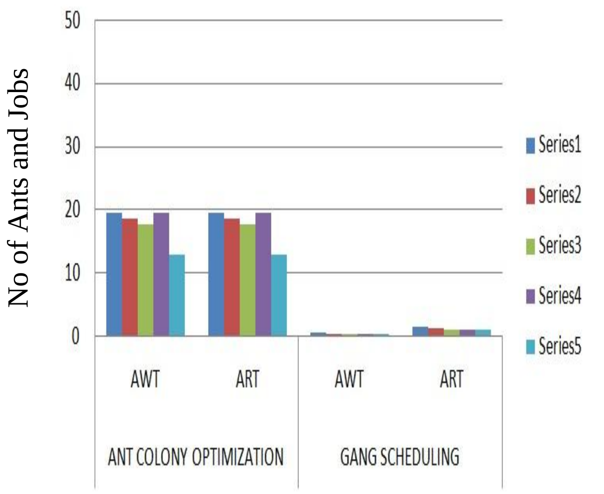


Job	WAITING TIME(WT) (ZERONEMT)	ZERONEMT (AWT)
10	5	0.5
20	9	0.45
30	11	0.36
40	18	0.45
50	19	0.38

IX. COMPARISION OF BOTH ANT AND GANG:

[Table 7. Average ACO Waiting Time and Response Time & Average
Gang Scheduler Waiting Time and Response Time]

NO OF JOBS AND ANTS	ANT COLONY OPTIMIZATION		GANG SCHEDULING	
	AWT	ART	AWT	ART
10	19.62	19.62	0.5	1.5
20	18.52	18.53	0.45	1.25
30	17.62	17.62	0.36	1
40	19.55	19.55	0.45	0.97
50	12.86	12.86	0.38	1



Waiting time and Response time

X. RESULTS:

The result arrives that ACO AND GANG SCHEDULING has been studied for calculating the waiting time and response time. So to check the performance of both algorithms performed some test based on scheduling the jobs in depth we come to know that Gang Scheduling outperforms better than the ACO. The job scheduling problems in grid environment and Cloud environment is solved using ACO and GANG SCHEDULING algorithm whereas GANG SCHEDULING has great potential, for scheduling the job according to there performance and utilization to the original structure are still necessary in order to significantly improve its performance. Gang scheduling is used to improve the convergence speed and diversity improvement for job allocation. Although this technique can be enhanced. And also Gang Scheduling can be used as an evolutionary framework for all types of job scheduling process and therefore gang scheduling is better than Ant Colony Optimization.

XII. CONCLUSION:

Grid computing aims to assign tasks to computing nodes and minimize the execution time of tasks as well as workload across all nodes. Despite of the intractability, the scheduling problem is of particular concern to both users and grid systems. The Gang Scheduling approach achieves optimal schedule than the traditional Ant colony algorithm. At the same time, the positive and negative feedback are applied to avoid stagnation situation encountered in searching. Here the modified pheromone updating is allocating the resources and obtain the shortest path optimally and adaptively in scalable, dynamic and distributed environment. But when compared to Gang Scheduling it groups the related threads scheduled as a unit. All members of Gang run in side by side manner on different time shared CPUs. This enhancement process has been achieved optimal scheduling by completing the tasks with minimum execution time as well as utilizing the resource in an efficient way.

XIII. REFRENCES:

[1] Yongsheng Hao, Guanfeng Liu, Rongtao Hou,Yongsheng Zhu, Junwen Lu, “Performance Analysis of Gang Scheduling in a Grid”, Springer Science, April 2014.

[2] Ioannis A, “Performance and Cost evaluation of Gang Scheduling in a Cloud Computing System with Job Migrations and Starvation Handling”, IEEE 2011.

[3] Y. Zhangy H. Frankez J. E. Moreirazy, “An Integrated Approach to Parallel Scheduling Using Gang-Scheduling, Backfilling and Migration”, IEEE Transactions on industrial informatics, Vol. 10, pp. 11, November 2013.

[4] M. Dorigo and T. St¨utzle, Ant Colony Optimization. MIT Press, 2004.

[5] H.Li, and C.Peng Lam, “Software Test Data Generation Using Ant Colony Optimization,” In Transactions on Engineering, Computing and Technology, 2005.

[6] P.Zhao, P.Zhao, and X.Zhang, “New Ant Colony Optimization for the Knapsack Problem,” in the Proceedings of the 7th International Conference on Computer-Aided Industrial Design and Conceptual Design, Nov 2006, pp 1-3.

[7] H. LI , C. Peng LAM, “An Ant Colony Optimization Approach to Test Sequence Generation for Statebased Software Testing,” In the Proceedings of the Fifth International Conference on Quality Software, September 2005, pp.255-264.

[8] X. Chen, Q. Gu, A. Li, D. Chen, ”Building Prioritized Pairwise Interaction Test Suites with Ant Colony Optimization,” In the Proceedings of Ninth International Conference on Quality Software [QSIC], 2009, pp. 347-352.

[9] P. R. Srivastava, and V. K. Rai ,”An ant colony optimization approach to test sequence generation for control flow based software testing,” In the Proceedings of 3rd International Conference on Information Systems, Technology and Management (ICISTM‘09), Springer Berlin Heidelberg, March 2009, pp. 345-346.

[10] Etsion Y. and Feitelson D. G., User-level communication in a system with gang scheduling. 15th Intl. Parallel & Distributed Processing Symp., Apr 2001.

[11] sowmya suryadevera, jaishri chourasia, sonam rathore & abdul jhummarwala load balancing in computational grids using ant colony optimization algorithm. international journal of computer & communication technology (ijcct) issn (online): 2231 - 0371 issn (print): 0975 –7449 vol-3, iss-3, 2012.

[12] http://en.wikipedia.org/wiki/Gang_scheduling.

[13] S. Umarani, Efficient Multiple Ant Colony Algorithm for Job Scheduling In Grid Environment / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3(2), 2012,3388-3393.

[14] Luís Fabrício Wanderley Góes1, Carlos Augusto Paiva da Silva Martins, “Reconfigurable Gang Scheduling Algorithm”.