# Modified Ant Colony Algorithm for Grid Scheduling

Mr. P.Mathiyalagan [1], S.Suriya[2], Dr.S.N.Sivanandam[3]

[1]Lecturer, Department of Computer Science and Engineering, PSG College of Technology, Coimbatore.
[2] Post Graduate Student Department of Computer Science And Engineering, PSG College of Technology, Coimbatore.
[3]Professor and Head Department of Computer Science and Engineering,
PSG College of Technology, Coimbatore.

**Abstract : The grid computing system is a new, powerful and innovative system for a group of heterogeneous distributed computing systems. It requires grid scheduling to achieve high performance. The efficient scheduling of independent jobs in a heterogeneous computing environment is an important problem in domains such as grid computing. In general, finding optimal schedule for such an environment using the traditional sequential method is an NP-hard problem whereas heuristic approaches will provide near optimal solutions for complex problems. The Ant colony algorithm, which is one of the heuristic algorithms, suits well for the grid scheduling environment using stigmeric communication. The proposed Ant colony algorithm in this paper has a modified pheromone updating rule which solves the grid scheduling problem effectively than that of the existing Ant colony algorithm.**

***Keywords: Scheduling, Heuristic approach, Pheromone, Stigmery***

## I. INTRODUCTION

Grid is defined as a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed autonomous resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements. Grid technology is defined as the technology that enables resource virtualization, on-demand provisioning and resource sharing between organizations. It can be confined to the network of computer workstations within a corporation or it can be a public collaboration.

The term Grid computing originated computer power as easy to access as an electric power grid. Grid computing is the process of applying the resources of many computers in a network to a single problem at the situations like - usually to a scientific or technical problem that requires a great number of computer processing cycles or access to large amounts of data. It involves the use of software that can divide and form out pieces of a program to as many as several thousand computers. It looks as distributed and large-scale cluster computing and as a form of network-distributed parallel processing.

Grid computing is an emerging computing model that provides the ability to perform higher throughput computing by taking advantage of man networked computers. They use the resources of many separate computers connected by a network to solve large-scale computation problems. They provide the ability

- To perform computations on large data sets, by breaking them down into many smaller ones.
- To perform many more computations.

Grid computing appears to be a promising trend for three reasons:

1. Its ability to make more cost-effective use of a given amount of computer resources.
2. A way to solve problems that cannot be approached without an enormous amount of computing power.
3. It suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as collaboration towards a common objective.

Scheduling is a key concept in computer multitasking operating systems, multiprocessing operating system design and in real-time operating system design. In modern operating systems, there are typically many more processes running than there are CPUs available to run them. Scheduling refers to the way processes are assigned to run on the available CPUs. This assignment is carried out by software known as a scheduler or is sometimes referred to as a dispatcher.

Grid Scheduling – Mapping of jobs to resources as per their requirements and properties. It is proposed to solve large complex problems. Grid

scheduling is an intelligent algorithm capable of finding the optimal resource for processing a job. Major function is to find the optimal resources and to allocate them to each individual job. The grid scheduler is shown in figure1. The objectives of a grid scheduler are overcoming heterogeneousness of computing resources, maximizing overall system performance, such as high resource, utilization rate and supporting various computing intensive applications, such as batch jobs and parallel applications. The grid scheduler is mainly concerned with the following: CPU utilization - to keep the CPU as busy as possible, throughput - number of process that complete their execution per time unit , turnaround time - amount of time to execute a particular process and response time - amount of time it takes from when a request was submitted until the first response is produced.

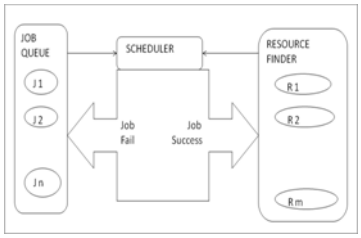


Figure 1: Grid scheduler Architecture with its Components

The term heuristic is used for algorithms which find solutions among all possible ones. These algorithms, usually find a solution close to the best one and they find it fast and easily. The ant colony optimization algorithm (ACO), is a heuristic algorithm for solving computational problems which can be reduced to finding good paths through graphs. This algorithm is a member of ant colony algorithms family, in swarm intelligence methods. Initially proposed by Marco Dorigo in 1992 in his PhD thesis. It is the first algorithm was aiming to search for an optimal path in a graph; based on the behavior of ants seeking a path between their colony and a source of food. The original idea has since diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behavior of ants.

## II. LITERATURE SURVEY

Inspired by the remarkable ability of the social insects to solve the problems, Dorigo[1] introduced highly creative new technological design principles for seeking optimized solutions to extremely difficult real-world problems, such as network routing and task scheduling. Thus an interface was set up between biology and technology fascinating. He describes the transaction of observed ant behavior into working optimization algorithms. The optimization process is characterized by a positive feedback loop, where the probability with which an ant chooses a path increases with the number of ants that previously chose the same path. It is termed as an Autocatalytic optimizing Process by Macro Dorigo[2]. A "single-ant" autocatalytic process usually converges very quickly to a bad suboptimal solution. Luckily, the interaction of many autocatalytic processes can lead to rapid convergence to a subspace of the solution space that contains many good solutions, causing the search activity to find quickly a very good solution, without getting stuck in it. In other words, all the ants converge not to a single solution, but to a subspace of solutions; thereafter they go on searching for improvements of the best found solution. The three algorithms defined show a different sensitivity to the parameters. Ant-density shows for $\beta$ a monotonic decrease of the tour length up to $\beta$=10. After this value the average length starts to increase. The tests on the other parameters show that $\alpha$ has an optimum around 1 that $\rho$ should be set as high as possible. The initial assumption, made by M.Dorigo [3], is that the amount of pheromone on a branch is proportional to the number of ants that used the branch in the past. This assumption implies that pheromone evaporation is not taken into consideration. In this model, the probability of choosing a branch at a certain time depends on the total amount of pheromone on the branch, which in turn is proportional to the number of ants that are used in the branch until that time. The stigmeric communication is achieved but there is no formal proof of the pheromone driven shortest path finding behavior in the general case is missing. The contribution done by Zhihong, Xiangdan and Jizhou [4], focuses on inherent parallelism and scalability that make the algorithm suitable for grid computing task scheduling. The scalability of the ant algorithm is validated using the proposed simple grid simulation architecture for resource management and task scheduling. The results of the paper carried out by Hui Fan ,Zhen Hua[5], reveals a good mathematical model to solve the shortest path problem. The probability of selecting new point is determined by the distance and the pheromone of the two cities .The evaporation rate parameter "$\rho$" has no effective role in the mathematical model since "$\alpha$" is used in pheromone updating formula. The algorithm proposed in Stefka Fidanova and Mariya Durchova [6], focuses on guarantee of good load balancing on machines. The initial assumptions were $\rho$=0.5 which will produce an effective result. No role of $\alpha$ and $\beta$ in this paper, whereas $\alpha$ and $\beta$ reduces complexity. This methodology is much effective for large number of tasks than number of machines. The paper carried out

by Huiyan , Xue-Qin-Shein and Xing Li[7] focuses on a new algorithm which is based on the general ant adaptive scheduling heuristics and an added in load balancing factor , related to job finishing rate. It is introduced to alter the pheromone. This is done to make job finishing rates at different resources being similar and the ability of the systematic load balancing will be improved. The trail intensity will be changed from $\Delta\tau j$ to $\Delta\tau j+C\lambda j$ ($C>0$ is a coefficient of the load balancing factor). When more jobs are finished, the trail intensity increases and when the jobs are not completed, the trail intensity decreases. This method has addressed only the concern of computation and communication capability of network, it needs more deliberation. The paper on Super schedulers by Li Liu , Yi Yang and Wanbin Shi [8] has the following contribution towards grid scheduling. Superschedulers can work co-ordinated with each other. A superscheduler can submit the jobs to the other neighboring superschedulers when it hasn't the communicational capabilities to run the job or it reaches the maximum resource limit threshold like CPU busy, memory unavailable or diskfull. When a job requires a resource, the job will be submitted to one superscheduler within the same administrative domain. A job is said to found a solution only when it has been successfully allocated to a resource. After the jobs finds a solution, a path between the beginning superscheduler to which the job is submitted and the final superscheduler in which a resource is actually allocated to the job has been made. In case of computational grid, a set of jobs will concurrently find the solution. A job can be submitted to any one of the superschedulers in computational grid. But no details about the decision of which superscheduler the job is submitted to has not been discussed in this paper. Each of resource requiring jobs has an ant like agent. When a job is successfully allocated, the agent with it will deposit a pheromone on the path it travelled and it will affect the path the following job will choose. The pheromone deposited in computational grid is actually some numeric information. This numeric information considers the agent's current history or performance which can be read or written by any agent. Pheromone trails are the only communication channels among the agents. This stigmeric or stigmergetic form of communication plays a major role in the utilization of collective knowledge about the global resource information in the computational grid. All agents need an additional memory to memorize the superschedulers the job has visited. An evaporation mechanism must be introduced to the computational grid [8] which modifies pheromone information over time. Pheromone evaporation allows the agents to slowly forget its past history so that it can direct its search towards new direction without being over-constrained by past decisions. Pheromone evaporation is due to preventing the bad solutions to be made by the following agents as the result of scalability of the grid resources the resources can be temporarily be added to or removed from the grid and variability change their computational capabilities at runtime. The main objective of pheromone evaporation is to avoid stagnation, the situation in which all the jobs are allocated to same resource. The pheromone information database at superscheduler i [8] contains a decision table $A_i=[a_{ij}]$ of superscheduler i which is obtained by the composition of the local pheromone trail values with the local heuristic values. As to gain some insight into the influence of the parameters $\alpha$, $\beta$ and $\gamma$, the role of the parameters is the following. If $\alpha= 0$, the SuperSchedulers with less communication time and greater capabilities are more likely to be selected: this corresponds to a classical stochastic greedy selection. If $\beta = 0$, the communication time between SuperSchedulers is not considered: this is not reasonable while the grid is in global scope, which have to suffer from the communication cost. If $\gamma = 0$, the capability of the SuperScheduler is not considered which will lead to the bad decision when a SuperScheduler is in full workload but with less communication cost. E.M.Saad, M.El.Adawy, H.A.Keshk and Shahira .M [9] has modified the ant colony algorithms. The ant colony algorithm has a lot of control parameters that effect on the performance and quality of the algorithm. This paper has presented two modified transition equations like

(i) The exponential function is characterized by its large rate increase in values. This characteristic feature can be used to obtain the best result rapidly.

(ii) The logarithm function s characterized by converting a large range of values to a small range of values, so the two parameters $\alpha$ and $\beta$ are not required. The second modified equation uses $\log_{10}$(1+pheromone or visibility value at any path) , this due to the fact that $\log_{10}(x)$ is negative value when x is less than 1. It is very obvious, using any one of the above two equations will reduce the control parameter $\alpha$ and $\beta$ and in the same time, the numbers of iterations are reduced with increasing system performance.

This paper presented two modified transition equations that eliminate the effect of the two control parameters ($\alpha$, $\beta$) on the system performance, which result in decreasing the Algorithm execution time. The control parameters used in case of original ant colony algorithm were $\alpha= 1$, $\beta= 0.5$, $\rho=0.8$, $Q=5$. The control parameters used in case of first Modified ant colony algorithm were $\rho=0.8$, $Q=0.01$, and in case of second Modified ant colony algorithm were $\rho=0.8$,

Q=5. The results from all algorithms used are the same but the time to obtain that result from the two modified Ant colony Algorithms is 1/20 less than the time to obtain the same result with the original Ant colony Algorithm which is 1/4.5 less than the time to obtain the same result in the Genetic Algorithm. The results show that, as the task graph increase in size, the two modified Ant colony Algorithm produce best result (according to choose minimum number of processors and minimum scheduling time) compared to the Original Ant colony Algorithm and Genetic Algorithm. The paper by Siriluck Lorpunmanae, Mohd Noor Sap, Abdul Hanan Abdullah and Chai Chompoo- inwai [10] aims at the development of a general framework of grid scheduling using dynamic information and an ant colony optimization algorithm to improve the decision of scheduling. Grid computing is a heterogeneous environment, which is also the dynamic environment. The scheduled jobs rarely coincide with or between actual execution time and the expected in case of real computing environment. The two main challenges of job scheduling are

- In grid system – no one has the ability to fully control all jobs.
- In dynamic resources – the difference between the expected execution time with the actual time (in algorithm).

The proposed approach is to develop scheduling algorithm with the objective to minimize total tardiness time of jobs based on ACO. The completion time of job $j^{th}$ in machine it is given as $C_{ij}=a_j+r_j+P_{i,j.}$. The tardiness of the jth job in machine i is given as $T_{i,j}=\max(C_{ij}-d_{j,0})$. The heuristic desirability of assignment job j on machine i directly moves to machine m , is inversely proportional to the completion time of job that has been assigned on machine as $n_j(i,m)=1/(a_j+r_j+P_{m,j})$. It is notable that the different performance and number of machines do have significant impact on schedule lengths. Again these three authors joined together to extend their concept over Multi constraints in their next paper [11], which proposes two kinds of pheromone and three kinds of heuristic information to guide the search direction of ants. Each ant will make use of the above information based on their probabilities. The information of partial solutions are used to modify the bias of the ants so that inferior choices will be ignored. The three kinds of different heuristic information to guide search direction of ants are duration greedy, cost greedy and overall greedy. The paper by Kousalya.K and Balasubramanie.P [12] deals with resource management and scheduling. The resources are heterogeneous in terms of architecture, power, configuration and availability. This complicates the task scheduling problem. The major objective of grid scheduling is to reduce the makespan. Hence the scheduling must consider some specific characteristics of the job and decide the metrics to be used accordingly. Ant algorithm, which is one of the heuristic algorithm suits well for the grid scheduling environment. This paper proposes a modified ant algorithm for Grid scheduling problem that is combined with local search. The proposed Ant colony algorithm takes into consideration the free time of the resources and the execution time of the jobs to achieve better resource utilization and better scheduling. In the evaluation study, a number of intensive experiments are conducted using the standard bench mark problem. The result shows that the proposed Ant colony algorithm is capable of producing high quality scheduling of jobs to grid resources. Thus the algorithm can be used to design efficient dynamic schedulers for real time grid environments.

## III. ANT COLONY ALGORITHMS

Ant colony optimisation algorithms [Dorigo 1996] are multi-agent systems, which consist of agents with the collective behaviour (stigmergy) of ants for finding shortest paths.Ant colony algorithms were inspired by the observation of real ant colonies. Ants are social insects, that is, insects that live in colonies and whose behavior is directed more to the survival of the colony as a whole than to that of a single individual component of the colony. Social insects have captured the attention of many scientists because of the high structuration level their colonies can achieve, especially when compared to the relative simplicity of the colony's individuals.

An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find shortest paths between food sources and their nest. While walking from food sources to the nest and vice versa, ants deposit on the ground a substance called pheromone, forming in this way a pheromone trail. When more paths are available from the nest to a food source, a colony of ants may be able to exploit the pheromone trails left by the individual ants to discover the shortest path from the nest to the food source and back. It is also interesting to note that ants can perform this specific behavior using a simple form of indirect communication mediated by pheromone laying, known as stigmergy.

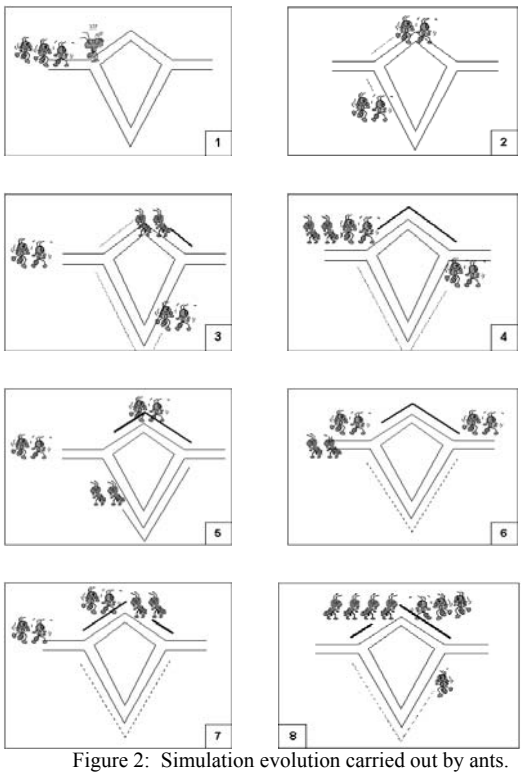


Figure 2: Simulation evolution carried out by ants.

The following experiment in figure 2 gives two paths to the food source, but one of them is twice longer than the other one. The ants again will start to move randomly and explore the ground. Probabilistically, 50% of the ants will take the short way while the 50% others will take the long way, as they have no clue to conclude the ground configuration. The ants taking the shorter path will reach the food source before the others and leave behind them the trail of pheromones. After reaching the food, they will turn back and try to find the nest. At the cross, one of the paths will contain pheromones although the other one will be not explored. Hence the ant which carries the food will take the path already explored, as it means it is the way to the nest. As the ant is choosing the shortest way and will continue to deposit pheromones, the path will therefore become more attractive for others.

The ants who took the long way will have more probability to come back using the shortest way, and after some time, they will converge toward using it. Consequently, the ants will find the shortest path by themselves, without having a global view of the ground. By taking decision at each cross according to the pheromones amount, they will manage to explore, find the food, and bring it back to the nest , in an optimized way.

A colony of ants begins with no solutions. Each ant constructs its own solution by making decisions, using existing problem constraints and heuristics combined with experience which is analogous to a substance called pheromone. The colony then reinforces decisions in the construction process according to their successes by adding pheromone, which also decays to mitigate against poorer decisions. The main purpose in this project is to investigate the ant colony algorithm technique as a means of constructing effective sequences of heuristic moves. In the same way, we can easily equip a colony of ants with initial solutions and let them collectively learn about the heuristic space, using this knowledge to guide their selection of appropriate solutions. Thus ant colony algorithm is used to find the optimal solution for scheduling tasks. The grid is composed of number of hosts. Each host has several computational resources. The resources may be homogeneous or heterogeneous. The grid scheduler finds out the better resource of a particular job and submits that job to the selected host. The grid scheduler does not have control over the resources and also on the submitted jobs. Any machine in grid can execute any job, but the execution time differs. The resources are dynamic in nature. As compared with the expected execution time, the actual time may be varied at the time of running the jobs to the allocated resource. The grid scheduler's aim is to allocate the jobs to the available nodes. The best match must be found from the list of available jobs to the list of available resources. The selection is based on the prediction of the computing power of the resource. So, lots of problems are needed to be solved in this area. The grid scheduler must allocate the jobs to the resources efficiently. The efficiency depends upon the criteria; one is makespan and the other is flow time. These two criteria are very much important in the grid system. In order to allow the ants to share information about good solutions a policy for updating the pheromone trail must be established. Allowing only the best ant to leave the pheromone after each iteration, makes the search much more aggressive and significantly improves the performance of ant colony optimization algorithms. This is the policy opted. Also, the best ant can be defined as either the iteration best ant or the best ant solution explored so far. In order to allow ants to forget poor information, each pheromone value is also decayed at this stage, this is implemented with a parameter $\rho$ which takes a value between 0 and 1 , if $\rho$ is set to one then no decay will take place , if $\rho$ is zero then each pheromone value will be wiped at each iteration and the pheromone trail is effectively switched off. Pheromone updating rule is given by:

$\tau_{ij}(t)new = [\ \tau_{ij}(t)old\ ] + [\ \{\rho\} * \Delta\tau_{ij}(t)\ ]$ ---(1)

Where

$\tau_{ij}(t)$ → Trail intensity of the edge(i,j).

$\rho$ →Evaporation rate.

$\Delta\tau_{ij}(t)$ →Additional pheromone when job moves from scheduler to resource.

The ants usually build a solution using both the information stored in the pheromone trail and the heuristic function. The ant solution building technique is an attempt to follow the concept of the best heuristic method. Each ant starts with an empty schedule and the processor $p_{ij}$ best which will complete each unscheduled job $j_1$, $j_2$, $j_3$…. $j_n$ earliest is established. A job j is then probabilistically chosen to schedule next based on the pheromone value between j and its best processor and heuristic value. The probability of selecting job j to schedule next is given by the following equation (2). In equation (2), $\alpha$ is a parameter which defines the relative weighting given to the pheromone information, and $\beta$ defines the relative weighting given to the heuristic information. If $\alpha$ is set to zero then only heuristic information is used and the ants effectively perform a probabilistic search. If $\beta$ is set to zero only pheromone information is used.

The probability selection is given as

$$P_{ij}(t)^k = [\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta} / \Sigma_{u \in Allowed(k)} \tau_{iu}(t)]^{\alpha} * [\eta_{iu}(t)]^{\beta} \quad \ldots\ldots\ldots\ldots (2)$$

Where

$P_{ij}(t)$ →Probability to move along the path (i→j).
$\tau_{ij}(t)]$ →Trail intensity of the edge(i,j).
$\eta_{ij}(t)$ → Visibility (1 / $distance_{ij}$ ).

The chosen job is then allocated to the best selected ant of each iteration. This process is repeated until all jobs have been scheduled and a complete solution has been built. Each ant in the colony builds a solution in this manner in each iteration. Once all the ants have built a solution the pheromone trail update procedure is performed as described above. It was observed in the test runs that the ants often take some time to start building good solutions because it takes a few iterations before the pheromone trail is populated with good job-processor pairings. After that, ant systems where algorithmically enunciated for optimization in problems like the salesman traveller and others. Ants are social beings with high structured colonies based on very simple individual behavior. Ants smell pheromone and when choosing their way, they tend in probability to the paths marked with stronger pheromone concentrations. When the time pass the pheromone concentration decrease. Repeating same behavior they compose optimized trails that are dynamically defining and they use to find food sources and their nest. The historic algorithm was enunciated by Dr.Dorigo for salesman traveller. This environment is very similar to the Grid and can be used in a very direct way that we shown bellow, following the algorithm:

1. procedure ACO
2. begin
3. Initialize the pheromone
4. While stopping criterion not satisfied do
5. Position each ant in a starting node
6. Repeat
7. for each ant do
8. Chose next node by applying the state transition rate
9. end for
10. until every ant has build a solution
11. Update the pheromone
12. end while
13. end

III Pseudo code for Existing Ant colony Algorithm

## IV. PROPOSED MODIFIED ANT COLONY ALGORITHM

The proposed ant colony optimization is used to solve large complex problems. It requires grid scheduling to achieve high performance. Scheduling of independent jobs remains as a complex problem in grid environment. Hence better scheduling in grid systems can be achieved using heuristic approaches. The Ant colony algorithm – one of the popular heuristic approaches can be used. The basic Ant algorithm involves Transition Probability and Pheromone Updating Rule. Improved ant colony algorithm is, modified ant colony algorithm , used to achieve better scheduling to improve the performance of grid system. The modified ant colony algorithm has changed the basic Pheromone updating rule of original ant colony algorithm. The improved pheromone updating rule is given by :

$$\tau_{ij}(t)new=[\{(1-)/(1+\rho)\}*\tau_{ij}(t)old]+[\{\rho/(1+\rho)\}*\Delta\tau_{ij}(t)] \quad \ldots\ldots\ldots\ldots\ldots\ldots (3)$$

Where

$\tau_{ij}(t)$ → Trail intensity of the edge(i,j).
$\rho$ → Evaporation rate.
$\Delta\tau_{ij}(t)$ → Additional pheromone when job moves from scheduler to resource.

The modified ant colony algorithm is as follows:

1. procedure Improved_ACO
2. begin
3. Initialize the pheromone
4. while stopping criterion not satisfied do
5. Position each ant in a starting node
6. Repeat
7. for each ant do
8. Chose next node by applying the state transition rate

$$P_{ij}(t)^k = [\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta} / \Sigma_{uCallowed(k)} \tau_{iu}(t)]^{\alpha} * [\eta_{iu}(t)]^{\beta}$$

9. end for
10. until every ant has build a solution
11. Update the pheromone

$$\tau_{ij}(t)new = [\{(1-)/(1+\rho)\} * \tau_{ij}(t)old] + [\{\rho/(1+\rho)\} * \Delta\tau_{ij}(t)]$$

12. end while
13. end

IV Pseudo code for Modified Ant Colony Algorithm

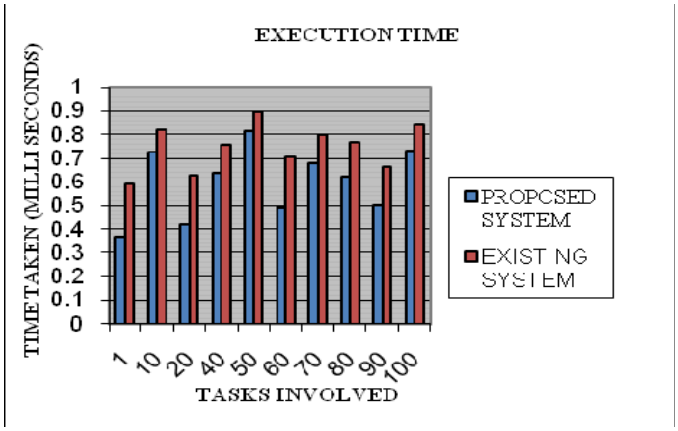
## V. EXPERIMENTAL RESULTS

The proposed Ant colony algorithm as a whole is a best suited method for tracking problem with large data sets. The above approach was simulated using GRIDSim toolkit and was found to be working efficiently and effectively. Experimental test carried out for a varied range of input set to ascertain the efficiency of the algorithm. From the results it is clearly evident that the proposed Ant colony algorithm offers better optimization a very fast rate.

| Number of tasks involved | Proposed Ant colony system (% of time taken for execution ) | Existing Ant colony system (% of time taken for execution) |
|---|---|---|
| 10 | 0.32 | 0.59 |
| 20 | 0.72 | 0.81 |
| 30 | 0.42 | 0.62 |
| 40 | 0.63 | 0.75 |
| 50 | 0.81 | 0.89 |
| 60 | 0.49 | 0.70 |
| 70 | 0.67 | 0.80 |
| 80 | 0.62 | 0.76 |
| 90 | 0.50 | 0.66 |
| 100 | 0.73 | 0.84 |

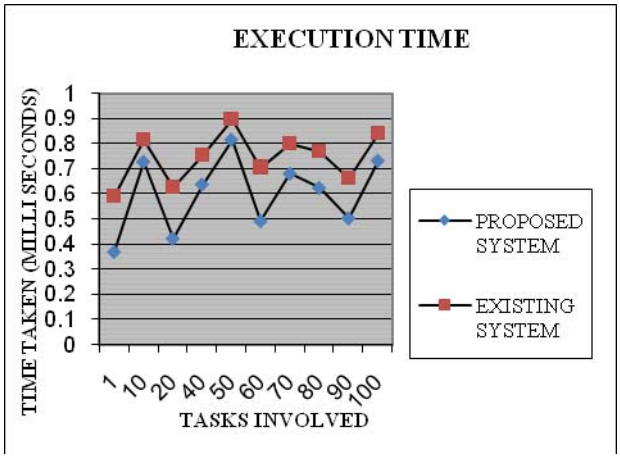
Table 1: Proposed System VS Existing System

The above table1 shows the amount of tasks considered for each period of execution. The results are tabulated for interval of every 10 tasks, starting from 10 tasks to 100 tasks respectively.




Execution time of Existing Ant Colony System VS Proposed Ant colony System

The above graph reveals that the proposed system works effectively than the existing system. The time taken to complete 10 tasks, 20 tasks, 30 tasks, 40 tasks, 50 tasks, 60 tasks, 70 tasks, 80 tasks, 90 tasks and 100 tasks are lesser when the proposed methodology is applied when compared to that of the existing system. Thus the proposed methodology is found evidently to work more effectively than that of the existing methodology.




Proportionate improvement of the Proposed Ant colony system VS Existing Ant colony system.

The above graph shows the level of improvement of working in the modified approach when compared to that of the existing approach. The curve obviously shows the improvement.

## VI. CONCLUSION

It has been convincingly proved in the recent research papers that task scheduling on computational grids is best solved by heuristic approach. The project tried to cover the state-of-the-art studies about one such heuristic namely Ant Colony Optimization (ACO) algorithm and its applicaion to grid systems.

The experimental results prove that the improved ant colony algorithm has effective role on grid scheduling. The modified pheromone updation rule makes the ant colony algorithm to work more efficiently than the original ant colony algorithm. Thus grid scheduling problems can be easily overcome using one of the heuristic approaches for optimization problems modified ant colony algorithms.

## REFERENCES

[1] M. Dorigo , V. Maniezzo and A. Colorni, ”AntSystem: An Autocatalytic Optimizing Process”,Technical Report 91-106 ,1991.

[2] Marco Dorigo and Gianni Di Caro , Luca M. Gambardella ,“Ant Algorithms for Discrete Optimization”, Research Paper, 1999.

[4] Zhihong Xu , Xiangdan Hou , Jizhou Sun ,” Ant Algorithm-Based Task Scheduling in Grid Computing ”, IEEE Conference Paper , May 2003.

[5] Hui Fan,Zhen Hua , Jin-Jiang Li, Da Yuan , “Solving a shortest path problem by ant algorithm”, Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2004.

[6] Stefka Fidanova and Mariya Durchova, “Ant Algorithm for Grid Scheduling Problem “, Research paper , 2005.

[7] Huiyan , Xue-Qin-Shein , Xing Li , Ming-Hui Wu, “An Improved Ant Algorithm for Job Scheduling in Grid Computing” , 18-21 August 2005.

[8] Li Liu, Yi Yang, Lian Li and Wanbin Shi , “Using Ant Colony Optimization for SuperScheduling in Computational Grid” , Proceedings of the 2006 IEEE Asia-Pacific Conference on Services Computing (APSCC'06) , 2006.

[9] E. M. Saad, M. El Adawy, H. A. Keshk, and Shahira M. Habashy , “The Ant Algorithm Modification” ,The 23rd National Radio Science Conference (NRSC 2006).

[10] Siriluck Lorpunmanae , Mohd Noor Sap , Abdul Hanan Abdullah and Chai Chompoo inwai , “An Ant Colony Optimization for Dynamic Job Scheduling in Grid Environment” , IEEE International Journal paper , 2007.

[11] Siriluck Lorpunmanae , Mohd Noor Sap , Abdul Hanan Abdullah and Aboamama Atahar Ahmed , “Multi-Constraint Dynamic Scheduling of Indepent jobs onto Grid Environment” , Jurnal Teknologi Maklumat,2007.

[12] Kousalya.K and Subramanie.P , “Ant algorithms for Grid Scheduling Powered by local search” , IEEE International Journal , 2008.