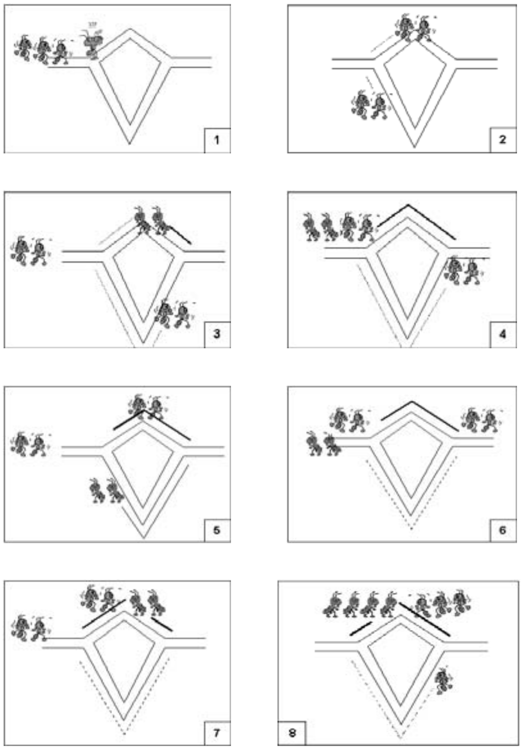


Figure 2: Simulation evolution carried out by ants.

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 ρ which takes a value between 0 and 1 , if ρ is set to one then no decay will take place , if ρ 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:

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

Where

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

ρ → 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 $j_1, j_2, j_3 \ldots j_n$ 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), α is a parameter which defines the relative weighting given to the pheromone information, and β defines the relative weighting given to the heuristic information. If α is set to zero then only heuristic information is used and the ants effectively perform a probabilistic search. If β 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 :

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

Where

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

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

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

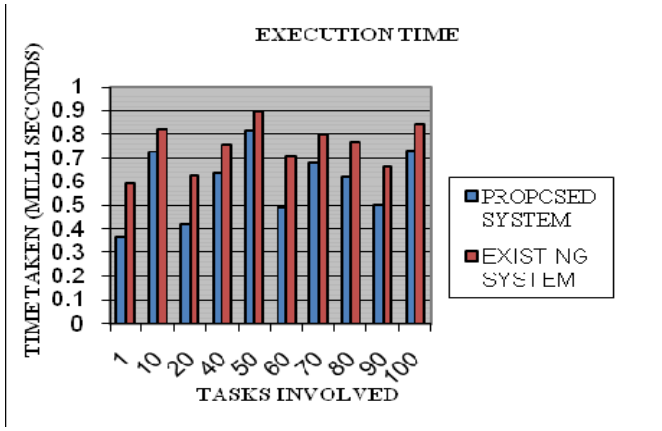
V. EXPERIMENTAL RESULTS

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

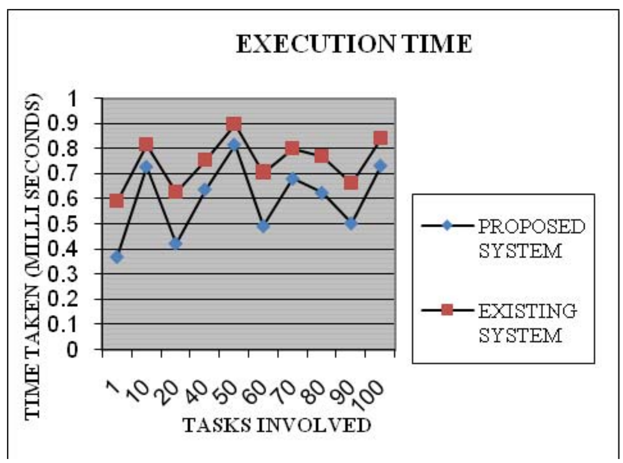
Table 1: Proposed System VS Existing System

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.