

GRID SCHEDULING USING ENHANCED ANT COLONY ALGORITHM

P. Mathiyalagan¹, U.R.Dhephthie² and S.N. Sivanandam³

Department of Computer Science and Engineering, PSG College of Technology, Tamil Nadu, India

E-mail: p_mathi2001@yahoo.co.in¹, dhephthie@yahoo.com²

³Akshaya College of Engineering and Technology, Tamil Nadu, India

Abstract

Grid computing is a high performance computing used to solve larger scale computational demands. Task scheduling is a major issue in grid computing systems. Scheduling of tasks is the NP hard problem. The heuristic approach provides optimal solution for NP hard problems. The ant colony algorithm provides optimal solution. The existing ant colony algorithm takes more time to schedule the tasks. In this paper ant colony algorithm improved by enhancing pheromone updating rule such that it schedules the tasks efficiently and better resource utilization. The simulation results prove that proposed method reduces the execution time of tasks compared to existing ant colony algorithm.

Keywords:

Pheromone, Swarm Intelligence, Inertia, Grid Scheduling

1. INTRODUCTION

Grid computing is the combination of computer resources from multiple administrative domains applied to achieve a goal, it is used to solve scientific, technical or business problem that requires a great number of processing cycles and needs large amounts of data.

Task scheduling is a key concept in computer multitasking and 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. 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. Sometimes these algorithms, they actually find the best solution, but the algorithm is still called heuristic until this best solution is proven to be the best. Some of the heuristic techniques are, Simulated Annealing, Tabu Search, Swarm intelligence which consists of two successful approaches of Particle Swarm Optimization (PSO) and Ant Colony Optimization algorithm (ACO).

The ACO algorithm is used because, the performance of ACO is better as compared to other approaches for various problems. Meta-heuristic algorithms are algorithms which, in order to escape from local optima, drive some basic heuristic; either a constructive heuristic starting from a null solution and adding elements to build a good complete solution and iteratively modifying some of its elements in order to achieve a better solution, Ant algorithm is a multi-agent approach to tackle difficult combinatorial optimization problems like the traveling salesman problem (TSP), the quadratic assignment problem (QAP), vehicle routing, graph coloring, and so on. In the ACO, meta-heuristic a colony of artificial ants cooperates in finding good solutions to difficult discrete optimization

problems. Cooperation is achieved through the pheromone trails deposited by the ants.

2. LITERATURE SURVEY

Ant Colony optimization (ACO) technique is employed in many optimization problems due to its simplicity and ability to tackle these problems successfully.

Marco Dorigo and Gianni Di Caro, Luca M. Gambardella [1], focus on the recent work on ant algorithms. This took inspiration from the observation of ant colonies foraging behavior, and introduces the ant colony optimization. It has been introduced for 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. A Stigmergy condition work successfully and produces better result and performance.

Zhihong, Xiangdan and Jizhou [2], proposed ant colony algorithm based task Scheduling in grid computing which focuses on inherent parallelism and scalability that make the algorithm suitable for grid scheduling. The scalability of the ant colony algorithm is validated using the proposed simple grid simulation architecture for resource management and task scheduling. The algorithm involves large set of parameters.

Stefka Fidanova and Mariya Durchova [3], which focuses on guarantee of good load balancing on machines Effective results are produced only when $\tau_0 = 0.01$ and $\rho = 0.5$ (initial assumptions). No role of α and β in this paper, whereas α and β reduces complexity. This methodology is much effective for large number of tasks than number of machines.

Huiyan, Xue-Qin-Shein and Xing L I [4] 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. 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.

Li Liu, Yi Yang and Wanbin Shi [5] proposed super schedulers can work co-ordinate with each other. A super scheduler can submit the jobs to the other neighboring super schedulers 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 disk full. When a job requires a resource, the job will be submitted to one super scheduler within the same administrative domain. A job is said to found a solution only when it has been successfully allocated to a resource. Produces heavy degradation of performance, when

$\alpha = 0$, $\beta = 0$ and $\gamma = 0$ the communication time was not considered in the above cases.

E. M. Saad, M. El Adawy, H. A. Keshk, and Shahira M. Habashy [6], proposed the ant colony algorithm modification. The enhanced ant colony algorithm has two enhanced transition equations that eliminate the effect of the two control parameters. The main merits of this modification results in decrease of the execution time. The algorithm yields best results and performance for minimum number of processors and scheduling time.

3. ACO ALGORITHM

Ant Colony optimization (ACO) is an algorithm modeled on swarm intelligence, and it constitutes some Meta heuristic optimizations. The algorithm was initially proposed by Marco Dorigo in 1992.

The ACO algorithm is a probabilistic technique for solving computational problems, which can be reduced to finding good paths through graphs. Ant 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. 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. Ants can smell pheromone and, when choosing their way, they tend to choose, in probability paths marked by strong pheromone concentrations. The pheromone trail allows the ants to find their way back to the food source (or to the nest). Also, it can be used by other ants to find the location of the food sources found by their nest mates. It has been shown experimentally that this pheromone trail following behavior can give rise, once employed by a colony of ants, to the emergence of shortest paths. That is, 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. In ACO algorithms a finite size colony of artificial ants with the above characteristics collectively searches for good quality solutions to the optimization problem under consideration. Each ant builds a solution, or a component of it, starting from an initial state selected according to some problem dependent criteria. While building its own solution, each ant collects information on the problem characteristics and on its own performance, and uses this information to modify the representation of the problem, as seen by the other ants. Ants can act concurrently and independently, showing a cooperative behavior. They communicate using artificial pheromone trails. An incremental constructive approach is used by the ants to search for a feasible solution. A solution is expressed as a minimum time (shortest path) through the states of the problem in accordance with the problem's constraints. The complexity of each ant is such that even a single ant is able to find a solution. High quality solutions are only found as the emergent result of the global cooperation among all the agents of the colony concurrently building different solutions. Accordingly to the assigned notion of neighborhood, each ant builds a solution by moving through a finite sequence of neighbor states. Moves are selected by applying a stochastic local search policy directed by pheromone

trail. The problem –specific heuristic information, and the knowledge, coded in the pheromone trails, accumulated by all the ants from the beginning of the search process is a key to successive cooperation among the ants. The decisions about when the ants should release pheromone on the environment and how much pheromone should be deposited depend on the characteristics of the problem and on the design of the implantation. Ants can release pheromone while building the solution, or after a solution has been built, moving back to all visited states, or both.

The amount of pheromone deposited is made proportional to the goodness of the solution an ant has built.

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

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_{ji} best which will complete each unscheduled job j_1, \dots, 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 Eq. (2). Transition probability is given by,

$$P_{ij}(t) = \{ [\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta \} / \{ \sum ([\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta) \} \quad (2)$$

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}).
- α is a parameter to control the influence of $\tau_{i,j}$
- β is a parameter to control the influence of $\eta_{i,j}$

Where, α and β are set to 1.

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.

4. PROPOSED METHODOLOGY

The enhanced ant algorithm has changed the basic pheromone updating rule of original ant algorithm. The improved pheromone updating rule is given by

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

Where,

- $\tau_{ij}(t)$ → Trail intensity of the edge (i,j).
- ρ → Evaporation rate (0.4).
- $\Delta \tau_{ij}(t)$ → Additional pheromone when job moves from scheduler to resource (0.2).

Pseudocode for Enhanced Ant Colony Algorithm:

1. Initialization set the pheromone, ρ and $\Delta \tau_{ij}(t)$.
2. Position each ant in the starting node

3. For each ant calculate the next node for traversal based on transition probability calculation.

Apply the Eq 1,

$$4. P_{ij}(t) = \{ [\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta \} / \{ \sum ([\tau_{ij}(t)]^\alpha * [\eta_{ij}(t)]^\beta) \}$$

5. Repeat step 3 until every ant has build a solution.

6. update the pheromone level on the path traversed based on the pheromone updation rule

Apply the Eq 3,

$$7. \tau_{ij}(t)_{new} = [\{ \rho + (1 - \rho / 1 + \rho) \} * \tau_{ij}(t)_{old}] + [\{ \rho - (\rho / 1 + \rho) \} * \Delta \tau_{ij}(t)]$$

8. Stopping criterion.

The enhanced ant algorithm has changed the basic Pheromone updating rule of original ant algorithm.

The improved pheromone updating rule is given by

$$\tau_{ij}(t)_{new} = [\{ \rho + (1 - \rho / 1 + \rho) \} * \tau_{ij}(t)_{old}] + [\{ \rho - (\rho / 1 + \rho) \} * \Delta \tau_{ij}(t)] \tag{4}$$

Where,

- $\tau_{ij}(t)$ → Trail intensity of the edge(i,j).
- ρ → Evaporation rate (0.4).
- $\Delta \tau_{ij}(t)$ → Additional pheromone when job moves from scheduler to resource (0.2).

The parameters of ant colony algorithm are $\alpha = 1$, $\beta = 1$ and $\rho = 0.4$. The table 1 shows the pheromone values of enhanced ant colony algorithm is very high compared to existing ant colony algorithm.

Table.1. Existing System VS Proposed System of Pheromone values in ACO algorithm

Nodes	Existing Method Ant colony algorithm	Proposed Ant colony algorithm
1	0.1426	0.1884
2	0.1180	0.1787
3	0.1075	0.1707
4	0.1030	0.1641
5	0.1010	0.1586

5. EXPERIMENTAL RESULTS

The enhanced ant colony algorithm was simulated by using ALEA toolkit. ALEA toolkit is a suitable tool used for scheduling of tasks in computational grid. Design of a grid environment which contains 14 resources and 100 tasks. The experimental results given in table 2 proves that proposed

method reduces the execution time of tasks 10,20,40,60,80 and 100 tasks compared to existing ant colony algorithm in computational grid .

Table.2. Execution time of tasks using enhanced ant colony algorithm in computational grid

Number of Tasks	Number of Resources	Proposed method make span time (ms)
10	2	91
20	4	85
40	8	74
60	10	68
80	12	67
100	14	65

6. CONCLUSION

The task scheduling is the difficult problem in computational grid. The heuristic method addresses these issues. The enhanced pheromone updating rule makes the ant colony algorithm to work more efficiently than the existing ant colony algorithm. The proposed method utilizes the idle resources efficiently. The experimental result proves that scheduling of tasks efficiently and minimizes the execution time.

RERERENCES

- [1] Marco Dorigo and Gianni Di Caro, 1999, “Ant Algorithms for Discrete optimization” Artificial life, MIT press.
- [2] Zhihong Xu, Xiangdan Hou and Jizhou Sun, 4 – 7 May 2003, “Ant Algorithm-Based Task Scheduling in Grid Computing”, Electrical and Computer Engineering, IEEE CCECE, Vol.2, pp.1107-1110.
- [3] Stefka Fidanova and Mariya Durchova, 2006, “Ant Algorithm for Grid Scheduling Problem”, Large Scale Computing, Lecture Notes in Computer Science No. 3743, Springer, pp.405-412.
- [4] Huiyan, Xue-Qin-Shein, Xing Li, Ming-Hui Wu, August 2005, “An Improved Ant Algorithm for Job Scheduling in Grid Computing”, pp.18-21.
- [5] Li Liu, Yi Yang, Lian Li and Wanbin Shi , L. Liu, Y. Yang, L. Li, W. Shi, 2006, “Using ant colony optimization for super scheduling in computational grid”, IEEE Asia-Pacific Conference on Services Computing, APSCC’06, pp.539-545.
- [6] E. M. Saad, M. El Adawy, H. A. Keshk, and Shahira M. Habashy, 2006, “The Ant Algorithm Modification”, The 23rd National Radio Science Conference, NRSC’06.