DOI: 10.21917/ijsc.2010.0009

ENHANCED HYBRID PSO – ACO ALGORITHM FOR GRID SCHEDULING

P. Mathiyalagan[1], U.R. Dhepthie[2] and S.N. Sivanandam[3]
Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India
Email: p_mathi2001@yahoo.co.in[1], dhepthie@yahoo.com[2], pma@cse.psgtech.ac.in[3]

Abstract
Grid computing is a high performance computing environment to solve larger scale computational demands. Grid computing contains resource management, task scheduling, security problems, information management and so on. Task scheduling is a fundamental issue in achieving high performance in grid computing systems. A computational GRID is typically heterogeneous in the sense that it combines clusters of varying sizes, and different clusters typically contains processing elements with different level of performance. In this, heuristic approaches based on particle swarm optimization and ant colony optimization algorithms are adopted for solving task scheduling problems in grid environment. Particle Swarm Optimization (PSO) is one of the latest evolutionary optimization techniques by nature. It has the better ability of global searching and has been successfully applied to many areas such as, neural network training etc. Due to the linear decreasing of inertia weight in PSO the convergence rate becomes faster, which leads to the minimal makespan time when used for scheduling. To make the convergence rate faster, the PSO algorithm is improved by modifying the inertia parameter, such that it produces better performance and gives an optimized result. The ACO algorithm is improved by modifying the pheromone updating rule. ACO algorithm is hybridized with PSO algorithm for efficient result and better convergence in PSO 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. One of the main strategies of grid computing is using software to divide and apportion pieces of a program among several computers, sometimes up to many thousands.

Some advantages of Grid Computing are,

- Results can then be concatenated and analyzed upon job(s) completion.
- Much more efficient use of idle resources. Jobs can be farmed out to idle servers.
- Grid environments are much more modular and don't have single points of failure. If one of the servers/desktops within the grid fails there are plenty of other resources able to pick the load. Jobs can automatically restart if a failure occurs.
- A client will reside on each server which sends information back to the master telling it what type of availability or resources it has to complete incoming jobs.
- Jobs can be executed in parallel speeding performance. Grid environments are extremely well suited to run jobs that can be split into smaller chunks and run concurrently on many nodes. Using things like MPI will allow message passing to occur among computer resources.

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 can be accurate, that is 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), Etc.The PSO algorithm is used because; it belongs to the class of direct search methods used to find an optimal solution to an objective function in a search space. Direct search methods are usually derivative-free, meaning that they depend only on the evaluation of the objective function. 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

Particle swarm optimization (PSO) technique is employed in many optimization and search problems due to its simplicity and ability to tackle these problems successfully.Kennedy and Eberhat developed a new evolutionary algorithm called particle swarm optimization (PSO).The PSO optimizes an objective function by iteratively improving a swarm of solution vectors, called particles, based on special management of memory. Each particle is modified by referring to the memory of individual swarm's best information. Due to the collective intelligence of these particles, the swarm is able to repeatedly improve its best observed solution and converges to an optimum. A heuristic approach proposed by Lei Zhang, Yuehui Chen, Bo Yang [1] based on particle swarm optimization is adapted to solving scheduling problem in the grid environment. Each particle is represented a possible solution. The approach aims to generate an optimal schedule so as to get the minimum makespan and maximum resource utilization while completing the tasks. The experimental results show that, the particle swarm optimization algorithm is more efficient and effective and provides better and optimized scheduling results than the Genetic algorithm. It is also found that the particle swarm optimization algorithm converges in the fast rate and it's suited for task scheduling. From the simulated experiment, the result of PSO algorithm is better than GA. Simulation results demonstrate that PSO algorithm can get better effect for a large scale optimization problem. Task scheduling based on PSO algorithm can be applied in the computational grid environment. The hybrid particle swarm optimization algorithm was proposed by M. Fikret Ercan [2] for the application of PSO in scheduling hybrid flow-shops with multiprocessor tasks. In order to improve the performance of PSO, hybrid techniques were employed. The experimental results show that the PSO and hybrid methods are more efficient and effective in scheduling basis. For the fixed processor case, the scheduling problem becomes more difficult to solve and results are relatively higher. However, in fix processor case, again the PSO-SA hybrid produced the best performance and results. Hybrid methods demonstrated significant performance improvement though it was at the expense of increased computational complexity. Particle swarm adaptation (James Kennedy) is an optimization paradigm that simulates the ability of human societies to process knowledge. The algorithm models the exploration of a problem space by a population of individuals; individuals' successes influence their searches and those of their peers. The algorithm is relevant to cognition, in particular the representation of schematic knowledge in neural networks. Particle Swarm Algorithm for Tasks Scheduling in Distributed Heterogeneous System was proposed by Xiaohong Kong, Jun Sun and Wenbo Xu [3]. A distributed heterogeneous system consists of a suite of processors or machines with different processing capacities. It can be performance-to-cost efficient to meet the diverse computation requirements if properly deployed. Task scheduling is a crucial issue to improve the efficiency of this architecture. It has been incorporated an efficient population-based search technique, Particle Swarm Optimization (PSO), with list scheduling and propose a hybrid PSO algorithm for tasks scheduling. The experiment results show that the proposed algorithm outperforms other algorithms in these aspects of performance and scalability. In practice, we encode the particle variables to the scheduling in order to utilizing the search capability of PSO. The major motivation of using PSO is theoretically find out an optimal scheduling list rapidly, then adopt a task duplication technique in the task allocation procedure. As a result Hybrid PSO results a better performance evaluation. The study of Brian Ivers ,Gary G.Yen [4] examines the optimization of the job shop scheduling (JSP) by a search space division scheme and use of the meta-heuristic method of particle swarm optimization (PSO) to solve it. The job shop scheduling problem (JSP) is a well known huge combinatorial problem from the field of deterministic scheduling. It is considered the one of the hardest in the class of NP hard problems. Particles are initialized in the search space of a particular problem by assigning them a position, which represents a solution to the objective function, and a velocity. The PSO algorithm is considered a very fast algorithm and is emerging as a widely studied used algorithm for optimization problems. Due to the memory characteristics of the PSO algorithm, it works better and there is no need to have the knowledge of other particles as, PSO takes care of it. Improvement of Particle Swarm Optimization Based on Neighborhood Cognizance and Swarm Decision was proposed by ZHU Meijie, LIU Hanxing, SUN Weiwei, ZHU TongLin [5]. The original PSO usually converges prematurely, and falls into the local optimal solution. Aimed at the shortcoming of PSO, here they have put forward an Improved PSO based on Neighborhood Cognizance (NCPSO) and Improved NCPSO based on Swarm Decision (SDNCPSO). These two improved PSO can reduce the possibility of converging prematurely. The results of experiment prove that these two improved PSO can improve the performance of global convergence in PSO and make PSO converge to global optimal solution faster. The parameters used here is not optimized, the effect of these parameters and optimized parameter setting should be analyzed and taken care.

This investigation [Shih-Tang Lo, Ruey-Maw Chen, Der-Fang Shiau and Chung-Lun Wu] introduced a particle swarm optimization (PSO) approach to solve the multi-processor resource-constrained scheduling problems [6]. There are two new rules are proposed and evaluated, named anti-inertia solution generation rule and bidirectional searching rule of PSO. In the job selection rule the solution generation is based on job priority. The anti-inertia search method is designed for job and processor without consider the inertia velocity. This mechanism can dynamic generate some new solution that ignores the determined velocity. In this study of bidirectional search rule, two classes of particles are used, forward and backward particles. They work on their own position and velocity, then the best solution found during the last iteration continues with the optimization process.The simulation results reveal that the proposed approach gives us the better and optimizes results. An Improved PSO Algorithm was proposed (BU Yan-ping, ZHOU Wei, YU Jin-shou) against the optimal objective of to minimize the total completing time [7]. This presents an improved particle swarm optimization (PSO) algorithm with discrete coding rule for grid scheduling problem. The discrete PSO algorithm was presented to solve grid scheduling problem within reasonable time. The algorithm is a very good optimizer for grid scheduling problem.

Ant Colony optimization (ACO) technique is employed in many optimization problems due to its simplicity and ability to tackle these problems successfully.This paper which was proposed by Marco Dorigo and Gianni Di Caro , Luca M. Gambardella [8], overviews 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. Stigmergy conditions work successfully and produce better result and performance.

Zhihong, Xiangdan and Jizhou [9], proposed "Ant Algorithm-Based Task Scheduling in Grid Computing" which 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 algorithm involves large set of parameters.

The , “Ant Algorithm for Grid Scheduling Problem” was proposed by Stefka Fidanova and Mariya Durchova [10], 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.

In the paper, “An Improved Ant Algorithm for Job Scheduling in Grid Computing” carried out by Huiyan , Xue-Qin-Shein and Xing L I [11] 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.

In the paper “Ant Colony Optimization for Super Scheduling in Grid” which was proposed by Li Liu , Yi Yang and Wanbin Shi [12] has the following contribution towards grid scheduling. 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 main part of Communication time was not considered in the above cases.

E. M. Saad, M. El Adawy, H. A. Keshk, and Shahira M. Habashy [13], proposed “The Ant Algorithm Modification” .The modified Ant algorithm has two modified 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.

Walid Elloumi, Nizar Rokbani and Adel. M.Alimi [14] developed a proposal to make PSO supervising an ant optimizer to solve optimization problems. the proposed algorithm can reduce the probability of being trapped in local optima and enhance the global search capability and accuracy. Pheromone deposit by the ants’ mechanisms would be used by the PSO as a weight of its particles ensuring a better global search strategy. This algorithm results in an efficient path.

3. PSO ALGORITHM

Particle swarm optimization (PSO) is an algorithm modeled on swarm intelligence, that finds a solution to an optimization problem in a search space, or model and predict social behavior in the presence of objectives.

The PSO is a stochastic, population-based computer algorithm modeled on swarm intelligence. Swarm intelligence is based on social-psychological principles and provides insights into social behavior, as well as contributing to engineering applications. The particle swarm optimization algorithm was first described in 1995 by James Kennedy and Russell C. Eberhart.

The particle swarm simulates this kind of social optimization. A problem is given, and some way to evaluate a proposed solution to it exists in the form of a fitness function. A communication structure or social network is also defined, assigning neighbors for each individual to interact with. Then a population of individuals defined as random guesses at the problem solutions is initialized. These individuals are candidate solutions. They are also known as the particles, hence the name particle swarm. An iterative process to improve these candidate solutions is set in motion. The particles iteratively evaluate the fitness of the candidate solutions and remember the location where they had their best success. The individual's best solution is called the particle best or the local best. Each particle makes this information available to their neighbors.

They are also able to see where their neighbors have had success. Movements through the search space are guided by these successes, with the population usually converging, by the end of a trial, on a problem solution better than that of non-swarm approach using the same methods.

Each particle represents a candidate solution to the optimization problem. The position of a particle is influenced by the best position visited by itself i.e. its own experience and the position of the best particle in its neighborhood i.e. the experience of neighboring particles. When the neighborhood of a particle is the entire swarm, the best position in the neighborhood is referred to as the global best particle, and the resulting algorithm is referred to as the gbest PSO. When smaller neighborhoods are used, the algorithm is generally referred to as the lbest PSO. The performance of each particle is measured using a fitness function that varies depending on the optimization problem.

Each Particle in the swarm is represented by the following characteristics:

- The current position of the particle
- The current velocity of the particle

The particle swarm optimization which is one of the latest evolutionary optimization techniques conducts searches uses a population of particles. Each particle corresponds to individual in evolutionary algorithm. Each particle has an updating position vector and updating velocity vector by moving through the problem space.

$$V_i^{k+1} = wV_i^k + c_1\, rand_1(\ )\ x\ (pbest_i - s_i^k) + c_2\ rand_2(\ )\ x\ (gbest - s_i^k) \quad (1)$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (2)$$

Where,

v_i^k is the velocity of i at iteration k, s_i^k is the current position of i at iteration k. c_1 and c_2 are positive constants and rand1 and rand2 are uniformly distributed random number in [0,1].The velocity vector is range of [-Vmax,Vmax].

In Velocity updating Eq (1), 3 terms that creates new velocity are,

- Inertia term, forces the particle to move in the same direction as before by adjusting the old velocity.
- Cognitive term (Personal best), forces the particle to go back to the previous best position.
- Social Learning term, forces the particle to move to the best previous position of its neighbors.

4. PROPOSED METHODOLOGY FOR PSO ALGORITHM

The inertia weight w weighting function in Eq (1), controls the momentum of the particle. The inertia weight can be dynamically varied by applying a scheme for the setting of the PSO, where w decreases over the whole run. The decrease depends on the start and end value of the weight given. A significant performance improvement is seen by varying the inertia term [15].

The Inertia term w, is provided with the below Eq (3) to make the convergence faster and easier.

$$w = w_{end} + (w_{start} - w_{end}) * \beta \quad (3)$$

where,

$\beta = (1 / 1 + (\alpha x / x_{\max})$

Parameters description,

w_{start}, Start value of the inertia weight (1.5)

w_{end}, end value of the inertia weight (0.3)

x, current iteration number

$x_{\max}$, maximum iteration number

α, used to manipulate the gradient of the decreasing factor, set to 5.

The inertia term should linearly decrease in order to facilitate exploitation over exploration in later states of the search.

5. 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 y 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.

$$\tau_{ij}(t)new = [\tau_{ij}(t)old] + [\{\rho\} * \Delta \tau_{ij}(t)] \quad (4)$$

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 pji best which will complete each unscheduled job j1,..,jn 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.

Transition probability is given by,

$$P_{ij}(t) = \{[\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta}\}/\{\Sigma([\tau_{ij}(t)]^{\alpha} * [\eta_{ij}(t)]^{\beta})\} \quad (5)$$

where

$P_{ij}(t)$ Probability to move along the path ($i \rightarrow j$).

$\tau_{ij}(t)$ 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.

6. PROPOSED METHODOLOGY FOR ACO ALGORITHM

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

$$\tau_{ij}(t)new = [\{\rho+(1-\rho/1+\rho)\}*\tau_{ij}(t)old]+[\{\rho-(\rho/1+\rho)\}*\Delta\tau_{ij}(t)] \quad (6)$$

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).

7. PARTICLE SOLUTION REPRESENTATION FOR GRID SCHEDULING

In grid environment, one of the most important issues is how to represent a solution for task scheduling. The solution representation ties up with the PSO algorithm performance. It's defined one particle as a possible solution in the population. The dimension n corresponding to n tasks and each dimension represents a task. The position vector of each particle makes transformation about the continuous position. The smallest position value, namely, (SPV) rule is used first to find a permutation corresponding to the continuous position x_i^k.

For the n tasks and m resource problem, each particle represents a reasonable scheduling scheme. The position vector x_i^k has a continuous set of values, where x_i^k is the position value of i particle with respect to the n dimension and s_i^k is the sequence of task of i particle in the processing order with respect to the n dimension.

Then the operation vector r_i^k is defined by following Equation, $R= s_i^k \bmod m$.

Table 1 illustrates the solution representation of particle x_i^k of PSO algorithm for 9 tasks and 3 processors. It's defined as that the start of sequence number is zero.

Table.1 Solution Representation

Dimension	x_i^k	s_i^k	r_i^k
0	3.01	5	2
1	7.96	8	2
2	-0.91	0	0
3	0.78	2	2
4	-0.31	1	1
5	1.85	4	1
6	5.26	7	1
7	4.75	6	0
8	1.77	3	0

8. HYBRID PSO – ACO ALGORITHM PROCEDURE

1. Begin
2. Initialize parameters;
3. Initialize population randomly;
4. Initialize each particle position Vector and velocity vector;
5. Evaluate the fitness, for each Particle using the ACO algorithm and find the personal best and the Global best;
6. While stopping criterion not satisfied do
7. Update each particle's velocity and Position;
8. Evaluate the fitness, for each Particle using the ACO algorithm and update the personal best and the global best;
9. End while
10. End

9. EXPERIMENTAL RESULTS

The experimental results show that the modified Particle swarm optimization algorithm and Ant colony optimization algorithm are able to get better schedule than the normal existing PSO and ACO algorithm in computational grid within an optimized time.

Table.2 Existing System vs Proposed System in PSO algorithm

Iteration	Existing method $e^{-(\alpha x/x_{max})}$	Proposed Method $(1 / 1 + (\alpha x / x_{max}))$
1	1.4992	1.4992
2	1.4988	1.4985
3	1.4982	1.4977
4	1.4976	1.4970
5	1.4970	1.4962

Table.3 Existing System vs Proposed System of Pheromone values in ACO algorithm

Nodes	Existing Method	Proposed Method
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

The Table 2 and Table 3 shows the level of improvement of working in the modified approach when compared to that of the existing approaches in PSO and ACO algorithms. The values in the Table 4 show the improvement. The time taken to complete

20, 60, 70 tasks, is lesser when the proposed methodologies are applied in the above hybrid PSO-ACO algorithm procedure.

Table.4 Proposed hybrid algorithm Makespan time for various tasks

Number Of Tasks Involved	Number Of Resources Involved	Proposed System(Ms)
20	2	2.9573
30	15	3.4031
60	15	3.6636
70	10	3.0880
70	20	2.5483

10. CONCLUSION

In this paper, a scheduling algorithm based on PSO and ACO algorithms are proposed for task scheduling problem on computational grids. Each particle represents a feasible solution. This paper aims at generating an optimal schedule so as to complete the tasks in a minimum time as well as utilizing the resources in an efficient way. The performances of the proposed approaches are compared with the existing approaches.

REFERENCES

[1] Lei Zhang, Yuehui Chen, Bo Yang, 2006, "Task Scheduling Based on PSO Algorithm in Computational Grid", in Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications.

[2] M. Fikret Ercan, 2008, "A hybrid particle swarm optimization approach for scheduling flow-shops with multiprocessor tasks", in International Conference on Information Science and Security.

[3] Xiaohong Kong, Jun Sun and Wenbo Xu, 2006, "Particle Swarm Algorithm for Tasks Scheduling in Distributed Heterogeneous System", in Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications.

[4] Brian Ivers, Gary G.Yen, 2007, "Job Shop Optimization through Multiple Independent Particle Swarms" in IEEE Congress on Evolutionary Computation.

[5] ZHU Meijie, LIU Hanxing, SUN Weiwei, ZHU TongL in "Improvement of Particle Swarm Optimization Based on Neighborhood Cognizance and Swarm Decision" IEEE.

[6] Shih-Tang Lo, Ruey-Maw Chen, Der-Fang Shiau and Chung-Lun Wu, 2008, "Using Particle Swarm Optimization to Solve Resource-constrained Scheduling Problems" in IEEE Conference on Soft Computing in Industrial Applications.

[7] BU Yan-ping, ZHOU Wei, YU Jin-shou, 2008, "An Improved PSO Algorithm and Its Application to Grid Scheduling Problem" in International Symposium on Computer Science and Computational Technology.

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

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

[10] Stefka Fid nova and Mariya Durchova, 2005, "Ant Algorithm for Grid Scheduling Problem ", Research paper.

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

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

[13] 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).

[14] Walid Elloumi, Nizar Rokbani and Adel. M. Alimi, 2009, "Ant Supervised by PSO", 4th International Symposium on Computational Intelligence and Intelligent Informatics, pp.161-166.

[15] P. Mathiyalagan, U.R.Dhepthie, S.N.Sivanandam, 2010, "Grid scheduling using enhanced PSO algorithm", (IJCSE) International Journal on Computer Science and Engineering, ISSN : 0975-3397, Vol. 02, No. 02, pp.140-145.