Grid Scheduling using Improved Particle Swarm Optimization with Digital Pheromones

SarathChandar A P, Priyesh V, Doreen Hephzibah Miriam D

Abstract—Scheduling is one of the core steps to efficiently exploit the capabilities of emergent computational systems such as grid computing. Grid environment is a dynamic, heterogenous and unpredictable computing system which shares different services among various users. Because of heterogenous and dynamic nature of the grid, the methods used in traditional systems could not be applied to grid scheduling and therefore new methods should be designed to address this research problem. This paper represents the technique of particle swarm optimization with digital pheromones which hasimproved solution characteristics. The main objective of the proposed algorithm is to find a solution that generates an optimal schedule which minimizes the flowtime in grid environment.Simulations have been carried out using GridSim for the improved PSO algorithm. Experimental studies illustrates that the proposed methodology, Improved PSO with digital pheromones is more efficient and surpasses those of PSO algorithms for the grid scheduling problem.

Index Terms—Digital Pheromones, Flow time, Grid Computing, Particle Swarm Optimization, Task Scheduling

1 Introduction

A computational grid [6] is defined as a hardware and software infrastructure, which provides dependable, consistent, pervasive, and inexpensive access to computational resources existing on the network.One of the most critical issues within the grid environments is resource sharing. Sharing the resources on grid is not primarily concerned with file exchange but rather on direct access to the computers, software, data, and other resources, as it is required by a range of collaborative problem solving and resource brokering strategies emerging in industry, science, and engineering. Thus resource sharing is controlled by the Resource Management System (RMS) and defined by the resource providers and consumers. The resource providers and consumers identify what is shared, who is allowed to share, and the conditions under which the sharing occurs.

The main goal of grid computing is solving problems, which are complex and time consuming. This goal may be achieved using the processing power of the computers existing on the grid environment. When the RMS receives the service request from a user, it divides the submitted service task into a subtask which can be executed concurrently. The RMS assigns these subtasks to the available resources for parallel execution. After the resources finish the assingned subtasks, they return the results back to the RMS. Finally, the RMS integrates the received results into the entire output of the submitted tasks.

To achieve to the goal of computational grids and to maximize the utilization of the resources in a grid environment, it is important how to schedule subtasks among the resources, a reasonable scheduling algorithm must be adopted in order to get the minimum completion time. So, task scheduling that needs to be addressed is one of NP-Complete problems in grid environment.

The remainder of the paper is organized as follows. Section 2 lists out the related work. Section 3 defines the Grid Scheduling problem. Section 4 introduces particle swarm optimization with digital pheromones. In section 5 the proposed methodology has been described. Experiment settings and results are given in section 6 and secition 7 concludes the paper with future work.

2 RELATED WORKS

Heuristic optimization algorithm is widely used to solve a variety of NP-Complete problems. Abraham et al [1] and Braun et al [3]address the dynamic scheduling of jobs to the geographically distributed computing resources. They provide an introduction of computational grids followed by a brief description of the three nature's heuristics namely Genetic Algorithm (GA), Stimulated Annealing (SA) and Tabu Search (TS). Yang Gao et al [7] proposed two algorithm that uses the productive models to schedule jobs at both system level and application level. M Aggrawal et al [2] presented a Genetic Algorithm based scheduler. The proposed scheduler can be used in both intra-grid of a large organization and a research grid consisting of large clusters, connected to a high bandwidth dedicated network. Shanshan song et al proposed a new space – Genetic Algorithm (STGA) for trusted job scheduling in which they consider three security driven heuristic modes:secure, risky and f-risky. Lee Wang et al [13] developed a heuristic based approach to matching and scheduling in heterogenous computing environment. LinJianNing et al [8] proposed a scheduling-algorithm based on genetic algorithm (GA) .Lein Zhang et al [14] proposed a task scheduling algorithm based on PSO for grid computing and showed that PSO

- *SsrathChandar A P and Priyesh V are currently pursuing bachelor degree program in computer science engineering in Sri Venkateswara College of Engineering, Chennai, India.*
  E-mail: sarathcse2008@gmail.com
  priyesh@hotmail.co.in
- *Doreen Hephzibah Miriam D is currently working as an Assistant Professor Computer Science engineering in Sri Venkateswara College of Engineering, Chennai, India.*
  E-mail: doreenhm@gmail.com

is able to get better schedule than genetic algorithm. A new PSO method for improved design space exploration using digital pheromones is proposed by Kalivarapu et al in [9]. In this paper, we propose to use this improved PSO algorithm with digital pheromones to solve the problem in grid environment. Our experimental results show that the proposed algorithm, improved PSO is efficient for task scheduling in computational grid, which reduces the fitness value as task size increases.

3 Problem Definiton

A resource in the computational grid is something that is required to carry out an operation, for example, a processor used for data processing. The research management of computational grid is responsible for research discovery and allocation of a task to a particular resource. Usually it is easy to get the information about the ability to process data in the available resources. In this paper, we will discuss the problem that$n$ tasks work on m computing resources with an objective to minimize the completion time and utilize the resources effectively if the number of tasks is less than the number of resources in grid environment, the tasks can be allocated on the resources according to the first-come-first-serve rule. If the number of task is more than the number of resources, the allocation of tasks is to be made by using efficient scheduling schemes. In this paper, we consider the number of tasks is more than the computing resources, so that a unique task can not be assigned to different resources, which implies that the proposed methodology avoids job migration [8]. The RMS in the grid environment contains all the information about the available resources.

In this paper, the Expected Time to Compute (ETC) each job on each machine is computed from the task set defined by the user and CPU time is derived from the grid system. Determination of ETC values is a separate research problem, and the assumption of such ETC information is used as benchmark for scheduling problems. In [3] ETC matrices are used to estimate the required time for executing each job in each machine. An ETC matrix is an $n*m$ matrix in which n is the number of jobs and m is the number of machines. One row of the ETC matrix contains the estimated exection time for a given job on each machine. Similarly one column of the ETC matrix consists of the estimated execution time of a given machine for each job. Thus, for an arbitrary job J_j and an arbitrary machineM_i, ETC (J_j,M_i) is the estimated execution time of$J_j$on M_i.

To formulate the problem, define T_i, i=1,2,3,…n as n independent tasks permutation and R_j, j=1,2,3,…m as m computing resources. Suppose that the processing time $P_{i,j}$ for task i computing on j resource is known. The completion time *C(x)* represents the total cost time of completion.

The objective is to find an permutation matrix x = ($x_{i,j}$), with $x_{i,j}$ =0, which minimizes the total costs

$$C(x) = \sum_{i=1}^{m} \sum_{j=1}^{m} P_{i,j} * x_{i,j} \quad (1)$$

subject to

$$\sum_{i=1}^{m} x_{i,j} = 1, \forall j \in T \quad (2)$$

$$x_{i,j} \in 0,1, \forall i \in R, \forall j \in T \quad (3)$$

The minimal *C(x)* represents the total length of schedule **using** available resources. The scheduling constraints guarantee that each task is assigned to exactly one resource.

4 Particle Swarm Optimization with Digital Pheromones

PSOis a population-based zero-order optimization method [10] that exhibits several evolutionary characteristics similar to GAs and SA. PSO is based on a simplified model of the social behaviour exhibited by the swarming behaviour of insects, birds, and fish. In PSO, randomly generated particle swarm(collection of particles) propogates toward a global optimum in a series of iterations, based on the information provided by two members, the best position of a swarm member in its history trail(pBest) and the best position attained by all swarm members(gBest). Based on this information a basic PSO generates a velocity vector indicating the direction of the swarm movement and updates the location of the particles.

Kalivarapu et al in [9] found that there are two drawbacks in the PSO algorithm. The first drawback of this method is that the particle updates are influenced by a limited number of factors. At any instance, each swarm member is directed by only two candidates, pBest and gBest. Using these two candidates potentially impedes the desirable exploratory characteristics. A second drawback is that the method is initial-condition-dependent. These drawbacks drives to poor locations specified by pBest and gBest in the initial stages of optimization this can potentially offset the swarm from attaining the neighbourhood of the solution in the design space. Toovercome these drawbacks, Digital Pheromones are used along with PSO.

A.Digital pheromones

Pheromones are chemical scents produced by insects essentially as a means of communication in finding suitable food and nesting locations. When more number of insects travel on the same path, the pheromone trail becomes stronger which denotes the availability of needed foods. The digital pheromonesinspired by this concept are used to explore search space and leave a marker in potential regions, where future investigations would be useful. This would aid in speeding up the process of searching for optimum solution.

B.PSO and digital pheromones

Kalivarapu et al. found that the benefits of digital pheromones from swarm interlligence and the adaptive applications described earlier can be merged into PSO to improve design space exploration. By **implementing the digital pheromone strategy** it was observed that the solution characteristics of the basic PSO algorithm could be drastically improved.

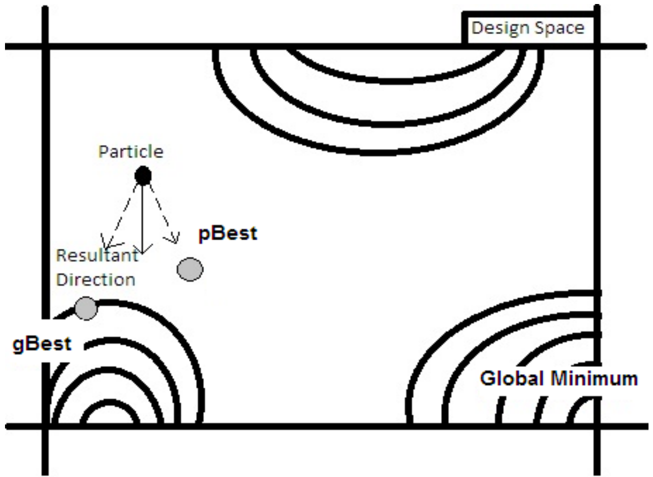

Figure 1. Particle movement within a basic PSO

In a basic PSO algorithm, the swarm movement is governed by the velocity vector. Each swarm member uses information from its previous best (pBest) and the best member in the entire swarm at any iteration (gBest). However it has been observed that the presence of pheromones in the design space would improve the solution characteristics by providing more information about the design space. This would be more useful when the information provided by pBestand gBest are insuffcent. Figure 1 displays the scenario of swarm member's.movement whose direction is guided by pBest and gBest alone [9]. If C_1>>C_2, the particle is strongly attracred to the pBest position. On the other hand if C_2>>C_1, the particle is strongly attracted to the gBest position. In this scenario dominated by C_2, as presented in Fig.1 neither pBest nor gBest leads to the swarm member to the global optimum, at the very least, not in this iteration adding additional computation to find the optimum.

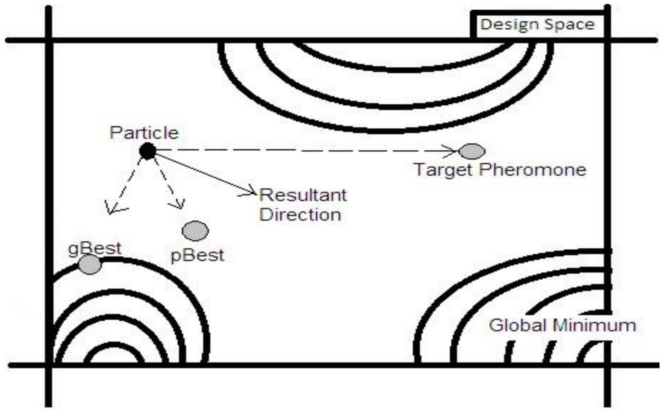

Figure 2. Particle movement within digital pheromones

Figure 2 shows the effect of impleting digital pheromones into the velocity vector [9]. In this implementation an additional velocity component is being added to velocity vector. An additional pheromone component potentially causes the swarm member to result in a direction different from the combined influence of pBest and gBest, thereby increasing the probability of finding the global optimum.

C.Digital pheromone implementation

Kalivarapu et al implemented the digital pheromones into the basic PSO algorithm. In this implementation the swarm is initialized like the basic PSO but 50 percent of the particles are selected randomly and made to release pheromones for the initial run alone. During the future runs, only those swarm members which experienced an improvement in the objective function were made to release the pheromones. The Pheromones from the current as well as the past iterations that are close to each other in terms of the design variable value can be merged into a new pheromone location, in order to manage the number of pheromones in the design space. In addition, the digital pheromones decay with iterations just as natural pheromones. Based on the current pheromone level and its position relative to a particle, a ranking process is used to select a target pheromone for each particle in swarm. This target position towards which a particle is to be attracted is added as an additional velocity vector component to pBest and gBest. This procedure is continued until a prescribed convergence criterion is satisfied.

This implementation is successfully used in [11] for the problem of personalized e-course composition, in which the improved PSO with DP outperformed the existing PSO approaches.

D. Determination of target pheromone

Kalivarapu et al found the need to identify a target pheromone for each of the particle owing to the fact a large number of pheromones was generated. They established a criterion which is a function of (a) the distance between the particle and the pheromoneand (b) the pheromone level. For each particle, target pheromone attraction factor P′ is computed to this effect, which is a product of the pheromone level (p) and the normalized distance between the particle and the pheromone (d). Equation (4) shows how the attraction factor P′ is computed, and (5) computes the distance between the pheromone and each particle in the swarm as guided by [9]

$$P' = (1\text{-}d)\,P \qquad (4)$$

$$d = \sqrt{\sum_{l}^{k}}\left(\frac{Xpk-Xk}{rangek}\right)^{2} (5)$$

Where k= 1: n, No. of design variables,
X_p= location of pheromone, X= location of particle

E. Velocity vector update

The velocity vector update, shown in (6), implements digital pheromones described. This involves a new component called the Target pheromone in the equation for velocity update.

$V_{i,d}^{t}$ = ω * $V_{i,d}^{t-1}$ + l + m + n (6)

l = C_i * $Rand_1$ * ($pbest_{i,d}$ - $Y_{i,d}^{t-1}$) (7)

m = C_2 * $Rand_2$ * ($gbest_{i,d}$- $Y_{i,d}^{t-1}$) (8)

n = C_3 * $Rand_3$ * ($Targetpheromone_{i,d} - Y_{i,d}^{t-1}$)(9)

Kalaivarapu et al observed that, C_3 is a user defined confidence parameter that combines the knowledge from the cognitive and social components of the velocity of a particle and complements their deficiencies. In a basic PSO, there is no memory of the path traversed by the swarm. According to the target phenomenon component was found to address the above cited issue. The target pheromone was found to stear the swarm towards the optimum solution by keeping track of the path visited by the particles. The additional pheromone term in the velocity vector update can considerably increase the computed velocity. Therefore, a move limit was applied to impose an upper bound on maximum value of the velocity vector. To ensure a fair amount of freedom in exploring the design space, the swarm is allowed to digressupto 10 percent of the range of the design variables initially. A decay factor of 0.95 is applied to this move limit in subsequent iteration as guided by which decreases the freedom to explore the design space as the iteration progress forward

5 Proposed Methodology

In the grid environment, scheduling is an NP Complete problem. Here, we propose an optimization technique to solve this problem.

A. Solution Representation

For grid scheduling algorithms, how to represent a solution is the challenging issue because solution representation ties up with the PSO algorithm performance. Each and every particle in the swarm represents one solution and the dimension *n* in the space corresponds to *n* number of tasks. The position vector of the particle is of continous value, $x_k^0 = [x_1^0, x_2^0, \dots, x_n^0]$. The continuous value is then transformed into discreet value $S_k^0 = [s_1^0, s_2^0, \dots, s_n^0]$using smallest position value rule (SPV Rule) [5]. Then the operation vector $r_k^0 = [r_1^0, r_2^0, \dots, r_n^0]$ is defined by the formula $R_k^i = S_k^i modm$. For example consider the solution representation of a particle in the algorithm for scheduling 8 tasks in 3 resources. We have m = 3 and n = 8. The solution representation is given by the following table.

From the Table I we can infer that zeroth task is assigned to second resource, third task is assigned to zeroth resource and so on.

Dimension	0	1	2	3	4	5	6	7
X_k^i	3.2	-0.5	-1.8	2.2	2.6	3.9	-0.9	4.6
S_k^i	5	2	0	3	4	6	1	7
R_k^i	2	2	0	0	1	0	1	1

Table I
SOLUTION REPRESENTATION

B. Improved PSO Algorithm with Digital Pheromone

The algorithm for grid scheduling using IPSO with digital pheromones is described. It consists of two phases: Initialization phase and Updation phase. ETC matrix (Expected Time to Compute Matrix) is given as input and the output of the algorithm is the operational vector R.

Algorithm1: Particle Swarm Optimization with Digital Pheromone for Grid Scheduling

```
Input: ETC[n][m],m – Number of Resources,nn – Number of Tasks
Output: Operation Vector R
Set k← 0
Set number of dimensions ← n
Generate m particles randomly
Generate initial velocities for m particles randomly
Select 50% of particles randomly and make them release pheromones
Apply SPV rule to find the discrete sequence S
Evaluate each particle using objective function
Set current position as best position for all particles
Find gBest
repeat
| Set k ←k + 1
| Decay the velocity vector
| Update the inertia weight
| Find the target pheromone
| Update velocity vector
| Update the position of the particles
| Find the sequence vector S by applying SPV
| Find resource vector R
|Update pBest
| Release pheromone if there is improvement in pBest
| Update gBest
Untilconvergence is satisfied;
```

1) *Intialization Phase*: Set $c_1 = c_2 = c_3 = 2$, w_0 = iter = 1. The initial population of particles is constructed randomly for PSO algorithm. The initialized continuous position values are$x_k^0 = [x_1^0, x_2^0, \dots, x_n^0]$is generated by the following formula.

$$x_k^0 = x_{min} + (x_{max} - x_{min}) * r \qquad (10)$$

Where x_{min} = -4.0, x_{max} = 4.0 and r is a uniform random number between 0 and 1.The initialized continuous velocity values, $v_k^0 = [v_1^0, v_2^0, \dots, v_n^0]$ is generated by the following formula.

$$v_k^0 = v_{min} + (v_{max} - v_{min}) * r \qquad (11)$$

Where v_{min} = -4.0, v_{max} = 4.0 and r is a uniform random number between 0 and 1.

We think the population size is the number of dimensions. 50% of the particles are randomly selected to release pheromones. Since X is the continuous valued variable, we need to convert it into discrete sequence. We apply SPV rule to find the permutation,$S_k^0 = [s_1^0, s_2^0, \dots, s_n^0]$. Apply the formula

$$R_k^i = S_k^i mod m \qquad (12)$$

To find the operation vector, $r_k^0 = [r_1^0, r_2^0, \dots, r_n^0]$. Now evaluate each particle in the swarm using the objective function. Find the best fitness value among the whole swarm and set it as global best value.

2) *Updation phase*: Update the iteration variable

$$iter = iter + 1 \qquad (13)$$

decay the pheromones if any. Update the inertia weight.

$$w^{iter} = w^{iter-1} * \beta \quad (14)$$

Find the target pheromone for each particle using (4). Apply (6) to update the velocity of the particle. Apply

$$x_{k+1} = x_k + v_{k+1} \quad (15)$$

to update the position of each particle. Apply SPV rule to find the permutation,$S_k^i = [s_1^i, s_2^i, \dots, s_n^i]$. Apply (12) to find the operation vector, $r_k^i = [r_1^0, r_2^0, \dots, r_n^0]$.

Each particle is evaluated by using operation vector to checkwhether the personal best has improved. If there is improvement, thenpBest pheromone is released. Update the global best.

Repeat the updation phase until the number of iteration steps exceeds the maximum number of steps.

6 Experiments

Various simulation studies were carried out to evaluate the performance of our scheduling algorithm using IPSO. We provide performance analysis comparing our proposed scheduling algorithm with the one using basic PSO. All the stimulations were carried out using GridSim [4] and the experimental setup and the results of the experiments are detailed below

A. Experimental Setup

In the basic PSO algorithm and IPSO algorithms, the value of the constants are set to 2 and the number of particles is equal to the number of tasks to be scheduled. In the IPSO implementation, the velocity vector is controlled by a factor of 0.95 in order to reduce the over-effect caused by extra target pheromone factor in the velocity vector update. Both the algorithm runs for the same instances of the problems. We represent data sets as x_y where x denotes the number of resources and y denotes the number of tasks.

B. Experimental Results

We simulated both PSO based scheduling and IPSO based scheduling in GridSim and compared the solution characteristics of both the algorithms. The fitness value is plotted against the number of iterations. From the first graph it is seen that IPSO produces much better solution charecteristics than basic PSO.

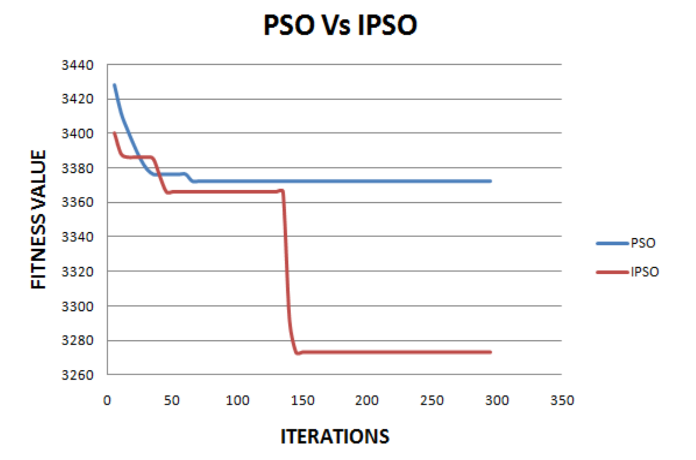

Figure 3.Comparison of solution characteristics of PSO and IPSO for 16_512

Then simulation for both the algortihm has been carried out by varying the number of resources and the number of tasks. For each and every x_y configuration, we run the simulation with 10 different input sets and the fitness values are averaged to find the fitness value. The fitness values obtained are plotted in second graph and it is found that better fitness values are generated as the data is scaled up. Thus the proposed scheduling algorithm is more scalable than the basic algorithm.

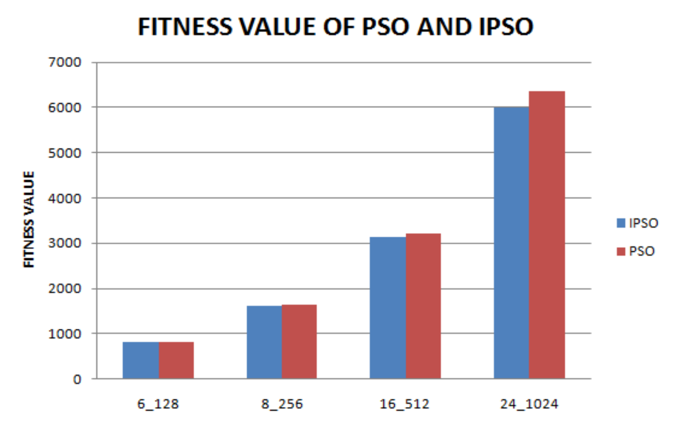

Figure 4.Comparison of scheduling PSO and IPSO.

We compute the improvement of the improved PSO algorithm over basic PSO as follows.

$$Improvement\ (\%) = \frac{\delta_1-\delta_2}{\delta_2} * 100\%(16)$$

Where δ_1 and $\delta_2$are the fitness values of two different algorithms.

No. of resources	No. of tasks	Improvement
6	128	1.84%
8	16	2.28%
16	512	2.71%
24	1024	5.87%

Table II
IMPROVEMENT IN ALGORITHM

The improvement of the algorithm for various input configurations has been listed in the table from the results it is clear that the IPSO based scheduler out-performs the PSO based scheduler as the number of jobs and resources increases. Thus our proposed method has greatest scalability than the existing method.

7. Conclusion and Future Work

In this paper scheduling algorithm based on Improved PSO with Digital Pheromones is proposed for task scheduling problem on computational grids. The flow times of the grid scheduling algorithms are compared and the proposed algorithm gave a good improvement as the number of tasks and resources are increasedwhich resulted in good scalability.

The proposed IPSO with Digital Pheromones algorithm is a key step of this research area. However, this paper mainly discusses the independent task scheduling. Our further studies will focus on multi objective grid scheduling and introducing the concept of satisfaction into concurrent tasks to design a task scheduling model based on agent technology and apply these methods to the development of grid platform.

REFERENCES

[1] *A. Abraham, R. Buyya, and B. Nath. Nature's heuristics for scheduling jobs on computational grids. The 8th IEEE International Conference on Advanced Computing and Communications, pages 45-52, 2000.*

[2] *M. Aggarwal, R. Kent, and A. Ngom. Genetic algorithm based scheduler for computational grids. 2005.*

[3] *T.D. Braun, H. J. Siegel, N. Beck, L. L. Boloni, M. Maheswaran, A. I. Reuther, J. P. Robertson, M. D. Theys, B. Yao, D. Hensgen, and R. F. Freund. A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogenous distributed computing systems. J. Parallel Distrib. Comput., 61:810-837, June 2001.*

[4] *R. Buyya and M. Murshed. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrency and Computation: Practice and Experience (CCPE), 14:1175-1220, 2002.*

[5] *M. FatihTasgetiren, Y.-C. Liang, M. Sevkli, and G.Gencyilmaz. Particle swarm optimization and differential evolution for single machine total weighted tardiness problem. International Journal of Production Research, 44:4737-4754, 2006.*

[6] *I. Foster and C. Kesselman. The grid 2 : Blueprint for a new computing infrastructure. Morgan Kaufmann, 2003.*

[7] *Y. Gao, H. Rong, and J. Huang. Adaptive grid job schedulings with genetic algorithms. Future Generation Computer Systems, 21:151-161, 2005.*

[8] *L. Jian-Ning and W. Hu-Zhong. Scheduling in grid computting environment based on genetic algorithm. Journal of computer research and development, 2004.*

[9] *V. Kalivarapu, J.-1.Foo, and E. Winer. Improving solution charaacteristics of particle swarm optimization using digital pheromones. Struct multidisc Optim, 37:415-427,2009*

[10] *J. Kennedy and R. Eberhart, particle swarm optimization. In proceedings of IEEE international conference of Neural Networks, 5:1942-1948, 1995*

[11] *A.P.SarathChandar, S. Dheeban, V.Deepak and S.Elias. Personalized e-course composition approach using digital pheromones in improved particle swarm optimization. In sixth international conference on Natural Computation, pages 2677-2681. IEEE, 2010.*

[12] *S. Song, Y. kwork, and K. Hwang. Security driven heuristics and a fast genetic algorithm for trusted grid job scheduling.2006.*

[13] *L. Wang, H. Seigel , V. Roychowdhury, and A. Maciejewski. Task matching and scheduling in heterogenous computing environments using a genetic algorithm based approach. Journal of parallel and distributed computing, 47:8-22, 1997.*

[14] *L. Zhang, Y. Chen, R. Sun and B. Jing, S ang Yang. A task scheduling algorithm based on pso grid computing. International journal of Computational Intelligence Reseach, 4:37-43, 2008*