

# LBMP SO Algorithm for Balancing the Load in Grid Environment

Sivakumar K

Assistant Professor

Dept. of CSE

JCT college of Engineering and Technology

Coimbatore, Tamil Nadu, India

**Abstract**—Grid computing is the collection of computer resources from multiple locations to reach a common goal. The grid can be thought of as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing in that grid computers have each node set to perform a different task/application. In this paper, a heuristic approach based on particle swarm optimization algorithm is adopted to solving task scheduling problem in grid environment. By reviewing through different author papers, it can be seen that particle swarm optimization gives better performance in terms of time, efficiency and balanced workload, etc. This approach aims to propose mathematical model multi-objective Load Balancing Mutation particle swarm optimization (LBMP SO) to schedule and allocate tasks to resource. LBMP SO considers two objective functions to minimize makespan and total cost.

**Keywords**—Grid computing, Scheduling, Particle swarm optimization.

## I. INTRODUCTION

Heterogeneous Computing (HC) systems consist of mixed group of machines, communication protocols and programming environments and offer a diversity of architectural capabilities that has different execution requirements. One of the key challenges of HC system is the task scheduling problem. In general, scheduling is concerned with distribution of limited resources to certain tasks to optimize few performance criterions, like the completion time, waiting time. Task assignment problems can be classified into two categories based on the types of tasks [1]: scheduling a meta-task composed of independent tasks with no data dependencies and assigning an application composed of tasks with precedence constraints.

PSO has been found to be robust and is successfully applied in solving nonlinear, no differentiable multi-modal problems quickly. It is still in its infancy. Many research works have mentioned application of PSO in task scheduling. PSO is most successful met heuristic for generations of optimal scheduling solutions. PSO scans over solution space during each iteration and accumulates global best and local best solutions. This section presents review of recent proposals which considered PSO in the field of task scheduling in cloud environment. Originally PSO was proposed where PSO was

proposed as an optimization tool. Two types of PSO namely, Discrete PSO and Continuous PSO versions were proposed. With several passes over the search space and updating local best and global best solutions during each pass, PSO performed much faster than ACO or GA. In [2] authors introduced the concept of inertia weight into the original PSO. With introduction of inertia weight PSO could converge even faster. Initially inertia weight was proposed to lie in the range [0.9, 1.2], which can improve performance of PSO. Different values of inertia allowed better control over solution search space. Higher values of inertia weight will result in overshooting the and lower values will trap search in definite area in search space. A Cost Aware Modified PSO (CA-PSO) was proposed in [3]. In [4] authors exploit PSO for optimizing overall tasks completion cost in a workflow and respecting the given deadline constraints. The proposed met heuristic approach based on PSO succeeds whereas IC-PCP fails to meet application's deadline. In comparison IC-PCP failed to meet deadline constraints as IC-PCP ignored VM boot time. Results prove that PSO performs better than current state-of-the-art algorithms.

Proposal considered deadline constraint. Proposal generates constraint makespan and performs cost evaluation for various workflows like Montage, Ligo etc. When compared to SCS, proposed algorithm is capable of generating better schedules and achieved cost optimization. In authors proposed mathematical model using a Load Balancing Mutation (balancing) Particle Swarm Optimization (LBMP SO) and considered reliability and availability as the objective parameters of proposals.

LBMP SO used an algorithm to generate schedule and allocation for Grid computing environment. Algorithm considered available resources for generation of schedule and allocation patterns. Basic PSO suffers from free VMs, allocation of more than one task to same VM, allocation of same tasks to multiple VMs and premature convergence. LBMP SO takes into account execution time, transmission time, make span, round trip time, transmission cost and load balancing between tasks and achieved reliability in task scheduling. Idea of LBMP SO is to reschedule failure tasks to available VM. LBMP SO performance was compared with standard PSO, random algorithm and

Longest Cloudlet to Fastest Processor (LCFP) algorithm to show that LBMPSTO can save in make span, execution time, round trip time, transmission cost.

## II. RELATED WORKS

Swarm intelligence is an important research topic based on the collective behavior of decentralized and self-organized systems in computational intelligence. It consists of a population which simulates the animals' behavior in the real world. Now there are many swarm intelligence optimization algorithms, such as genetic algorithms, particle swarm optimization, ant colony optimization, bee colony algorithm, differential evolution, fish-warm algorithm, etc. Due to the simple concept, easy implementation and quick convergence, PSO has gained much attention and been successfully applied in a variety of fields mainly for optimization problems. So far, as for the constrained optimization problems, relatively less work based on PSO can be found than those based on other EAs. Parsopoulos and Vrahatis [22] proposed a non-stationary multi-stage assignment penalty function method to transform the constrained problem to the unconstrained problem. Simulation results showed that PSO outperformed other EAs, but the design of the multistage assignment penalty function is too complex.

In the work of Hu and Eberhart [23], the initial swarm contains only feasible solutions and a strategy to preserve feasibility is employed. Motivated by multi-objective optimization techniques, Ray and Liew [24] proposed a swarm algorithm with a multilevel information sharing strategy to deal with constraints. In their work, a better performer list (BPL) is generated by a multilevel Pareto ranking scheme treating every constraint as an objective, while the particle which is not in the BPL gradually congregates round its closest neighbor in the BPL. In these swarm intelligence algorithms, the most popular and widely used algorithms for solving the TSP are GAs, ACO and PSO.

Clerc [25] develops several algorithm variants with those operations and methods and applies them to the asymmetric TSP instance br17.attp. In his algorithm the positions are defined as TSP tours represented in vectors of permutations of the  $|N|$  vertices of the graph correspondent to the considered instance. This approach was applied to tackle the real problem of finding out the best path for drilling operations [27].

Particle swarm optimization is presented to solve traveling salesman problem [26], where the authors have proposed the concept of swap operator and swap sequence, and redefined some operators on the basis of them. This paper has designed a special PSO, but the special PSO does not improve the updating formula, and the experiment results are worse than ours. Hendtlass [28] proposes the inclusion of a memory for the particles in order to improve diversity.

The memory of each particle is a list of solutions (target points) that can be used as an alternative for the current

local optimal point. There is a probability of choosing one of the points of the particle's memory instead of the current best point of the whole swarm Pgdb. The size of the memory list and the probability are new parameters added to the standard PSO algorithm. The algorithm is applied to the benchmark TSP instance burma14.

The results obtained with algorithmic versions with several parameter settings are compared with the results of an Ant Colony Optimization algorithm. Pang et al. [29] extends the work of Wang et al. [26]. Their algorithm alternates among the continuous and the discrete (permutation) space. In order to avoid premature convergence, Pang et al. [29] use a chaotic operator. constant  $D_{min}$  and the cost of the tour represented in the particle's position.

## III. PROPOSED SYSTEM

### A. LBMPSTO

Load balancing mutation PSO used to reschedule tasks that failure to schedule. PSO have two problems. First problem, tasks may failure to allocate to virtual machine. Second problem, task may allocate to more than one VM. In this phase solve the problems by reschedule wrong tasks and take in account load balancing of virtual machine. Solving these problems help to achieve reliability, users assert task executed without failure, minimize execution, minimize round trip time and improve other parameters.

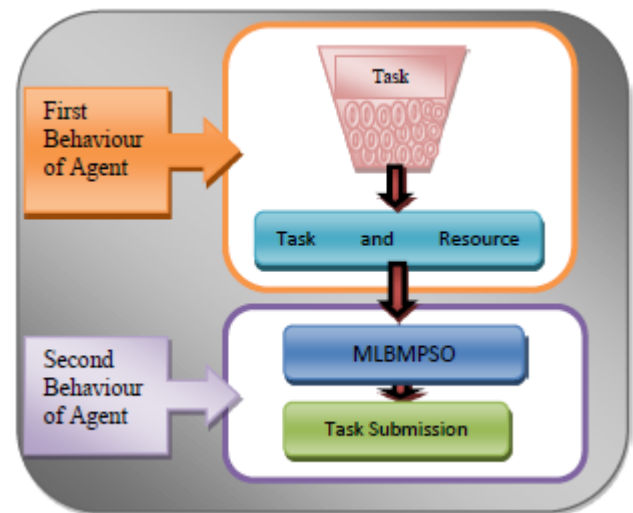


Fig.1: Proposed Model Structure

### B. Task Scheduling Problem Formulation of proposed systems

There are several tasks ( $t$ ) and several virtual machines ( $vms$ ). There are  $n$  tasks and  $m$  number of virtual machines. Each task may allocate to any  $vm$ . Fig.2 shows mapping of Tasks to virtual machines. Each task must schedule to only one virtual machine.

PSO attempts to select optimal distribution of tasks to virtual machines for achieving objective.

Three mathematical models proposed for task scheduling. Each model consists of objective function and several constraints. Objective function of first model is to minimize execution time based on expected execution time (EET<sub>ij</sub>) of task *i* in vm<sub>j</sub>. Equation (1) used in calculate processing time as: EET (processing time) = length<sub>i</sub> / mips<sub>j</sub>. length<sub>i</sub> is number of instruction of task *i* require to execute. mips<sub>j</sub> is number of Instructions executed by vm per second. Second objective function is to minimize transmission time (ETRT<sub>ij</sub>)(6). Expected transmission time ( ETRT<sub>ij</sub> ) of task *i* to vm *j* responsible for achieving second objective function. ETRT<sub>ij</sub> equals file size / bandwidth. To minimize round trip time (RTT) (3) is achieved by third mathematical model.

The RTT is the (latency) time for the whole procedure involving the sending and the receiving. ERTT<sub>ij</sub> is expected round trip time calculate by ETRT<sub>ij</sub> + delay + EET<sub>ij</sub> + delay. *x<sub>ij</sub>* is allocating task *i* to vm *j* or not . The value of *x<sub>ij</sub>* may one or zero. Each model has the same constraints. Each Task allocate to only one virtual machine achieve by first constraint in (2). Equation (3) and (4) represent resource of all virtual machine less than or equal resource of datacenter. *x<sub>ij</sub>* assign positive number (5).

Nomenclature	
N	The number of tasks
M	Number of virtual machines
<i>X<sub>ij</sub></i>	Decision variable of allocating task <i>i</i> to vm <i>j</i> or not
ERTT	Expected round trip time
mem <sub>j</sub>	Memory allocate to vm <i>j</i>
cpu <sub>j</sub>	Cpu allocate to vm <i>j</i>
Totalmem	Total memory of datacenter
Totalcpu	Total cpu of datacenter
W1	weight of first objective function
W2	weight of second objective function

#### Multi-objective task scheduling Mathematical Model

$$\text{Min } z = W_1 * \sum_{i=0}^n \sum_{j=0}^m \text{ERTT}_{ij} * x_{ij} + W_2 * \sum_{i=0}^n \sum_{j=0}^m \text{ETC}_{ij} * x_{ij} \quad (1)$$

Subject To:

$$\sum_{j=0}^m x_{ij} = 1 \quad \forall i \quad (2)$$

$$\sum_{j=0}^m \text{cpu}_j \leq \text{totalcpu} \quad (3)$$

$$\sum_{j=0}^m \text{mem}_j \leq \text{totalmem} \quad (4)$$

$$x_{ij} \geq 0 \quad \forall i, j \quad (5)$$

#### C. LOAD BALANCING MUTATION PARTICLE SWARM OPTIMIZATION (LBMP SO)

Obtaining an optimal schedule of tasks to resource with Considering constraints of a bi-objective optimization Problem are well-known problems in NP hard category[16]. The particle swarm optimization (PSO) algorithm is one of the heuristic techniques that used to obtain a feasible solution in reasonable time. PSO proposed. by Kennedy and Eberhart [17] . Initially, the PSO algorithm generates a set of particles randomly in the D dimensional search space. Particles defined as a potential solution to a problem. Each particle is represented by a Ddimensional vector *X<sub>i</sub>* where *i* ranges from 1 to *d* represent as (*x<sub>i1</sub>*, *x<sub>i2</sub>*, ..., *x<sub>id</sub>*).. Each particle is updated its position and its velocity according to equations 6, 7. In the iteration *t*, the velocity *v<sub>i</sub>(t)* has been update based on *v<sub>i</sub>(t-1)* is the velocity of the pervious iteration, *r<sub>1</sub>*, *r<sub>2</sub>* mean a uniform random variables between 0 and 1 this two random values are generated independently, *c<sub>1</sub>*, *c<sub>2</sub>* are a positive constant constants called acceleration coefficients, and *w* is the inertia weight. It also remembers the candidate solution of best fitness value it has achieved thus far during the search (individual best position (pbest)).

Also, the PSO algorithm maintains candidate solution of the best fitness value achieved among all particles in the swarm (global best position (gbest)). Equation (7) updates each particle's position using the computed *v<sub>i</sub>(t)*. Tasks allocated to vms using pso to achieve proposed mathematical model. In Task allocation using pso has some problem .such as, some task fail to allocate to vm or task allocate to more than one vm and premature convergence. Solving previous problem in Particle Swarm Optimization add Load balancing mutation to pso as show in Fig. 2. Load balancing mutation improved reliability,availability, minimize round trip time and minimize cost .The idea of Multi-objective Load balancing mutation Particle Swarm Optimization (MLBMP SO) reschedule the failure tasks to the available (VM) and reschedule tasks

that allocate to more than VM with take into account load of each vm. LBM guarantee all vm executed number of tasks appropriate with their load of vm. In LBM, First Determine failure tasks and calculate *load of virtual machines as load of vmi* = (resource of vmi /total resource)\*N. Then sort tasks based on resource needed and sort vms based on load. Last Reschedule failure tasks to vm based on load of each vm as in algorithm 1.

$$v_i^{k+1} = wv_i^k + c r \times \left( pbest_i - x_i^k \right) + c r \times \left( gbest - x_i^k \right) \quad (6)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (7)$$

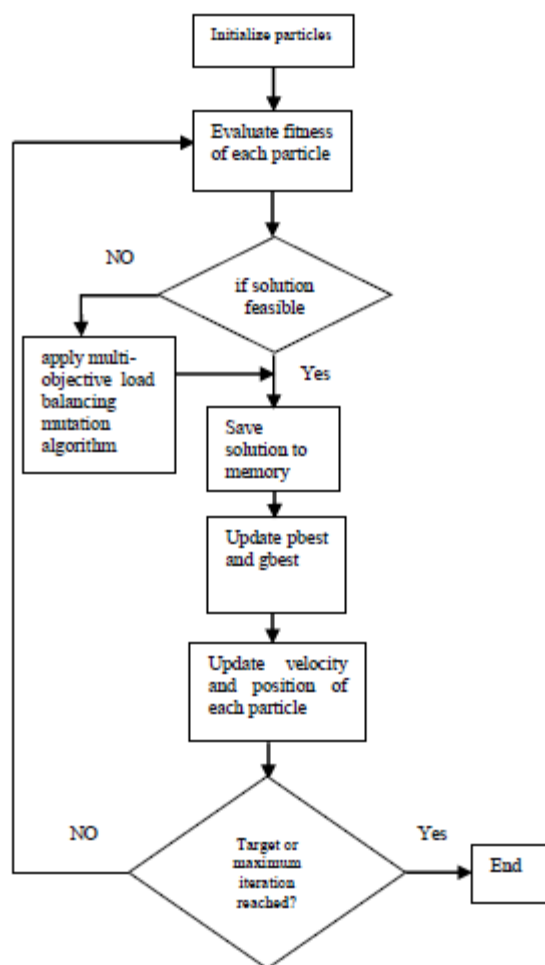


Fig.2: MLBMPSO Algorithm

#### Algorithm 1: Load Balancing Mutation Algorithm

```

Get best solution of pso
For all task { ti } ∈ T do
    Determine unallocated tasks
    Determine tasks allocated to more than one vm (wrong tasks)
End for
For all virtual machine { vmi } ∈ VM do
    Determine current tasks allocated to vmi (current load vm)
    Determine real load of vmi (real load vm)
End for
Sort vm based on real load
Sort wrong tasks based on resource needed
For all sorted virtual machine {svmi} ∈ VM do
    For all sorted task {sti} ∈ T do
        If real load vm > current load vm
            Schedule task from wrong tasks
            Remove task from sorted tasks list
            Current load vm++
        Else
            Break; // Exit to get next vm because this vm
                Take load based on resource
        End if
    End for
End for
  
```

## IV. SIMULATION RESULT AND EVALUATION

### A. GRIDSIM

Gridsim used to experiment proposed algorithm (MLBMPSO) and compared with, other algorithm. The experiments are implemented with 6 Datacenters with 50 VMs and 1000 tasks. The parameters of cloud simulation are shown in Table1.

### B.EXPERIMENTS AND RESULTS

In evaluating of scheduling heuristic, each task is independent to other task. The average cost and average make span are parameters used in comparison between two algorithms.

These comparisons obtained after 15 independent experiments done. We compared between Multi-objective load balancing mutation PSO, other algorithm [18].The result of comparisons between two algorithms based on different parameters when other algorithm group based on time show in fig.3-6.

In Figure5-6 show comparisons between two algorithms when other algorithm group based on cost. The conclusions show that MLBMPSO is best algorithm which improve availability, reliability and consider load balancing between virtual machines. Also, make span and cost.



Table 1 : parameters of cloud simulation

Parameters	Value
<b>Tasks(cloudlets)</b>	
Length of task	1000-20000
number of task	1000
file Size	1-500
output Size	1-500
<b>Virtual Machine</b>	
number of VMs	50
MIPS	500-2000
VM memory(RAM)	256-2048
Bandwidth	500-1000
<b>Datacenter</b>	
Number of Datacenter	6
Number of Host	3-6

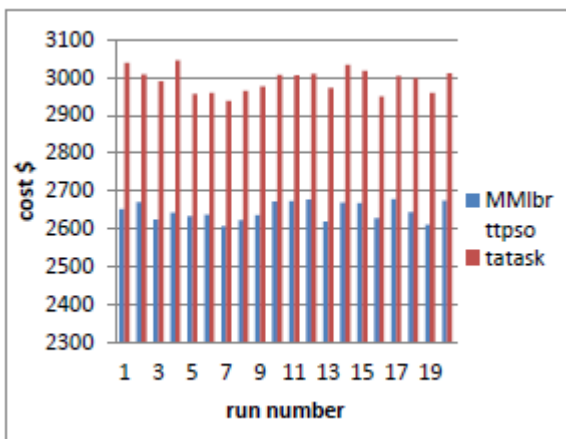


Fig.3: Average Cost

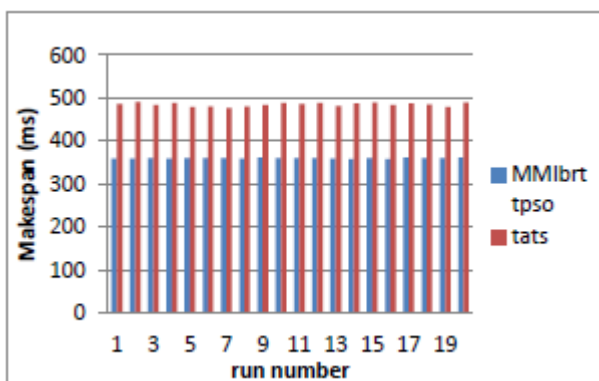


Fig.4: Average Makespan

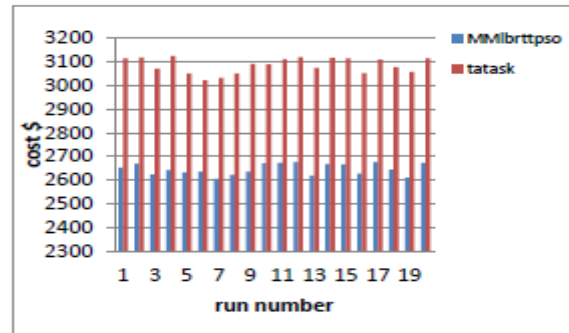


Fig.5: Average Cost

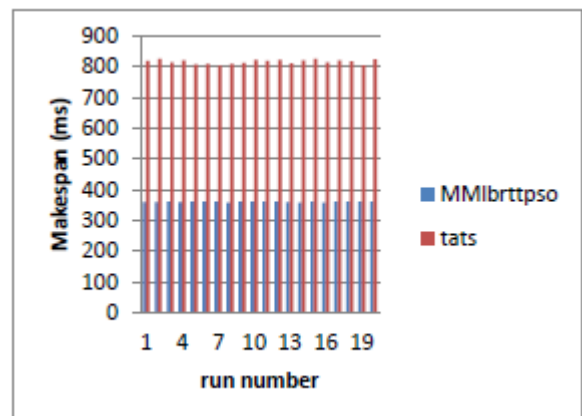


Fig.5: Average Makespan

## V. CONCLUSION

Task scheduling is one of the most important issues that effect in performance of Grid Computing environment.

There are many task scheduling algorithms which not consider important parameters such as availability and reliability. In this paper, we present Dynamic Multi objective task a scheduled based on load balancing Mutation Particle Swarm Optimization (MLBMPPO).

MLBMPPO improves the Reliability of Grid computing and consider availability of resources compared to other algorithms. MLBMPPO used to minimize total cost, minimize round trip time, improves task completion time, improve execution cost, good distribution of tasks onto resources, achieve load balancing between tasks and virtual machine and minimize the complexity in Grid computing environment. In addition, proposed algorithm consider the load balancing when schedule tasks to available and reschedule failure tasks to achieve reliability. It can be used to allocate any number of tasks and resources.

## REFERENCES

- [1] P. Mell, and T. Grance. (2011). The NIST Definition of Grid computing .NIST[Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [2] R.Buyya ,C.S. Yeo, S.Venugopal ,J.Broberg and I.Brandic “Grid computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5<sup>th</sup> utility” ,Future Generation Computer Systems , 2009 ,vol.25,no.6,pp.599–616.
- [3] Paul, M. et al. “Task scheduling in Grid computing using credit based assignment problem”, International journal of Comput. Sci. Eng, 2011, pp. 26-30.
- [4] A.I.Awad, N.A.El-Hefnawy and H.M.Abdel\_kader ,” Enhanced Particle Swarm Optimization For Task Scheduling In Grid computing Environments“ , in International Conference on Communication, Management and Information Technology (ICCMIT 2015),2015.
- [5] G.Caire, JADE Tutorial: JADE Programming for Beginners, <http://sharon.cselt.it/projects/jade/>
- [6] P. Angin and B. Bhargava, “An Agent-based Optimization Framework for Mobile-Grid computing”,Journal of Wireless Mobile Networks,Ubiquitous Computing, and Dependable Applications, 2013, vol. 4,pp. 1-17.
- [7] C.Mateos, E.Pacini and C. G.Garino,, “An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments”, Advances in Engineering Software, 2013, vol.56,pp. 38–50.
- [8] P.Samal and P.Mishra, "Analysis of variants in Round Robin Algorithms for load balancing in Grid computing", International Journal of Computer Science and Information Technologies, 2013, Vol. 4, no.3, pp. 416-419.
- [9] B.Mondala, K.Dasgupta and P.Duttab,” Load Balancing in Grid computing using Stochastic Hill Climbing-A Soft Computing Approach”, in 2nd International Conference on Computer, Communication, Control and Information Technology , on February 25 – 26, 2012, vol.4, PP. 783 – 789,.
- [10] S.Pandey, L.Wu, S.M.Guru2 and R.Buyya,” A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Grid computing Environments”, in IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010, pp. 400 - 407.
- [11] G.Liu, J.Li and J.Xu, “An Improved Min-Min Algorithm in Grid computing”, Proceedings of the International Conference of Modern Computer Science and Applications, 2012, vol .191,pp.47-52.
- [12] W.Wang, G.Zeng, D.Tang and J.Yao,”Cloud-DLS: Dynamic trusted scheduling for Grid computing”, Expert Systems with Applications , 2012,vol.39,no.3, PP. 2321– 2329,.
- [13] B. Xu, C.Zhao, E.Hu and B.Hu,”Job scheduling algorithm based on Berger model in cloud environment”, Advances in Engineering Software, 2011 ,vol. 42,no.7, PP. 419–425.
- [14] S.Ambike, D.Bhansali, J.Kshirsagar and J.Bansiwal “ An Optimistic Differentiated Job Scheduling System for Grid computing” International Journal of Engineering Research and Applications (IJERA) 2012, Vol. 2, no. 2, pp.1212-1214.
- [15] H.Zhong, K.Tao and X.Zhang ,” An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems “, in Fifth Annual China Grid Conference, ,on 16-18 July ,2010 , pp. 124 - 129.
- [16] A. S. Ajeena Beegom and M. S. Rajasree,” A Particle Swarm Optimization Based Pareto Optimal Task Scheduling in Grid computing “,in Proceedings of 5th International Conference, ICSI 2014, October 17-20, 2014 , pp 79-86.
- [17] Eberhart R , Kennedy J, “A New Optimizer Using Particle Swarm Theory”, 6th International Symposium. Micro Machine and Human Science, Nagoya;1995.p. 39-43.
- [18] M.Choudhary and S.K.Peddoju “A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment” ,International Journal of Engineering Research and Applications (IJERA), , 2012 Vol. 2, no.3, pp.2564-2568.
- [19] S.Ghanbaria and M.Othmana,” A Priority based Job Scheduling Algorithm in Grid computing”,in International Conference on Advances Science and Contemporary Engineering , 2012 ,vol.50, and PP. 778 – 785