Max-Min Particle Swarm Optimization Algorithm with Load Balancing for Distributed Task Scheduling on the Grid Environment

C.Kalpana[1] U.Karthick Kumar[2] and R.Gogulan[3]

[1]Assistant Professor, Department of Computer Science
V.L.B Janakiammal College of Arts and Science
Coimbatore, Tamilnadu, India

[2]Assistant Professor, Department of MCA & Software System
V.L.B Janakiammal College of Arts and Science
Coimbatore, Tamilnadu, India

[3]Research scholar,
School of Computer Sciences, Bharath University
Selaiyur, Chennai, Tamil Nadu, India -600 073

Abstract

In this paper proposes an algorithm namely max min Particle Swarm Optimization with load balancing techniques with the comparison of Swarm Intelligence Algorithms like Ant Colony Optimization. This algorithm is based on the task scheduling in grid environment. First of all we are calculating the QoS constraints to the PSO and ACO like, makespan, Cost and Deadline. Finally we had balanced the load to the particle swarm optimization, and compared with all other algorithms. We conclude the proposed algorithm is the best algorithm which is compared with ACO, PSO, Max PSO and Min PSO.

***Keywords**: Grid, Scheduling, load balancing, ACO, PSO.*

1. Introduction

The term “the Grid" was coined in the mid-1990s to denote a proposed distributed computing infrastructure for advanced science and engineering. Much progress has since been made on the construction of such an infrastructure and on its extension and application to commercial computing problems. Early defnitions for the Grid go back to 1998, when Carl Kesselman and Ian Foster dened the Grid as follows [1]: A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. Putting it in simple words, Grid computing aims to connect geographically distributed computers allowing their computational power and storage capabilities to be shared. With the rapid growth of data and computational needs, distributed systems and Grids are gaining more attention to solve the problem of large-scale computing [2]. There are several options for establishing distributed systems, and Grid Systems [3] are one of the common ones for distributed applications [2]. Various Grid application scenarios have been explored in both academia and industry. The rest of the paper is organized as follows: Section 2 presents Related work, Section 3 describes the Swarm Intelligence. In section 4 we describe the Basic PSO. In Section 5 discuss on the proposed work, load balancing problem will also defined clearly in section 6. Section 7 gives the Conclusion of this paper with future work.

2. Related Work

Ant colony optimization algorithms have been applied to many combinatorial optimization problems, ranging from quadratic assignment to fold protein or routing vehicles and a lot of derived methods have been adapted to dynamic problems in real variables, stochastic problems, multi targets and parallel implementations. It has also been used to produce near optimal solutions to the travelling salesman problem. K. Kousalya and P. Balasubramanie used task parcelling based Ant Algorithm for Grid Scheduling [12]. The performance of ACO algorithms for scheduling problems has been demonstrated in Ant colony optimization for resource constrained project scheduling [13]. Yarkhanan and Dongarra used simulated annealing for grid job scheduling. The Simulated Annealing scheduler is compared to an Ad Hoc Greedy scheduler used in earlier experiments. The Simulated Annealing scheduler exposes some assumptions built into the Ad Hoc scheduler and some problems with the Performance Model being used [14]. Abraham et al. used Tabu Search and Simulated Annealing for scheduling jobs on computational grids [15]. Genetic Algorithm for grid job scheduling is addressed in several works [16], [15]. The benefits of the usage of the Genetic Algorithms to improve the quality of the scheduling are discussed. The result of this paper suggests the usage of local search strategy to improve the convergence when the number of jobs to be considered is big as in real world operation [16]. Shih Tang Lo, et al., introduced a particle swarm optimization (PSO) approach to solve the multi processor resource constrained scheduling problems [17]. An improved PSO Algorithm was proposed by BU Yan-ping, et al., against the optimal objective to minimize the total completion time [18]. This presents an improved particle swarm optimization (PSO) algorithm with discrete coding rule for grid scheduling problem.

A heuristic approach proposed by Lei Zhang, et al., [17], based on particle swarm optimization is adapted to solving scheduling problem in the grid environment. Each particle represents a possible solution. The hybrid particle swarm optimization algorithm was proposed by M. Fikret Ercan [19], for the application of PSO in scheduling hybrid flowshops with multiprocessor tasks. In order to improve the performance of PSO, hybrid techniques were employed. Particle Swarm Algorithm for Tasks Scheduling in Distributed Heterogeneous System was proposed by Xiaohong Kong, Jun Sun and Wenbo Xu [13]. A distributed heterogeneous system consists of a suite of processors or machines with different processing capacities. Task scheduling is a crucial issue to improve the efficiency of this architecture. It simulates the process of a swarm of insects preying and works well in many global optimal problems [15]. In all these techniques, it is found that the resources are not balanced optimally. In this paper, an attempt is made to reschedule the tasks to the lightly loaded resources.

3. Swarm Intelligence

As artificial life and swarm intelligence techniques are increasingly being used for solving optimization problems, they have proven themselves as a good candidate in this area.

This can be inferred by recent research in the area.

The notion of complex collective behaviour emerging from the behaviour of many relatively simple units, and the interactions between them, is fundamental to the field of artificial life. The understanding of such systems offers new ideas in creating artificial systems which are controlled by such emergent collective behaviour; in particular, the exploitation of this concept might lead to completely new approaches for the management of distributed systems, such as load balancing in Grids [4]. Job scheduling is known to be NP-complete when we want to solve it on a single processor, therefore the use of heuristics and involving distribution is necessary in order to cope in practice with its dificulty [5].

Some of the algorithms are below,

a. Ant Colony Optimization
b. Particle Swarm Optimization
c. Intelligent Rain drops
d. Artificial Bee Colony

These all the algorithms which was included in the swarm intelligence techniques.

3.1 Ant Colony Optimization (ACO)

Ant colony optimization algorithm is one of the member of swarm intelligence. Here some metaheuristic optimizations are used to calculate the shortest path of every colony. This algorithm is initially proposed by Marco Dorigo in 1992 in his PhD thesis. In this algorithm mainly based on the behavior of ants seeking a path between their colony and a source of food [6].

3.2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization is a global optimization algorithm, based on the swarm intelligence. This algorithm can deal the problems which is the best solution is represented as a point or surface in an n-dimensional space. Initial Velocity can be calculate for the particular particle will move their path known. As well as the communication channel can send to another particle which is moving on the communication [7].

3.3 Intelligent Rain drops (IWD)

Intelligent Rain drops algorithm is swarm based nature inspired optimization algorithm, which has been inspired by natural rivers and how they find almost optimal paths to their destination. These near optimal or optimal paths follow from actions and reactions occuring among the water drops and the water drops with their riverbeds [8].

3.4 Artificial Bee Colony (ABC)

Artificial Bee Colony algorithm is a swarm ased meta-heuristic algorithm. This algorithm is introduced ny Karaboga in 2005. This algorithm simulates the foraging behaviour of honey bees. This algorithm have three phases. There are employee bee, onlooker bee and acout bee.in the employed bee and the onlooker bee phases, bees exploit the sources by local searchers in the neighbourhood of the solutions selected based on deterministic selection in the employed bee phase and the probabilistic selection in the onlooker bee phase.in the scout bee phase which is an analogy of abandoning exhausted food sources in the foraging proccess, solutions that are not beneficial anymore for search progress are abandoned, and new solutions are inserted instead of them to explore new regions in the search space. The algorithm has a well-balanced exploration and exploitation ability [9].

4. BASIC PSO

PSO has been developed through simulation of simplified social models. The features of the method are as follows,(i)The method is based on researches about swarms such as fish schooling and a flock of birds. (ii)It is based on a simple concept. Therefore, the computation time is short and it requires less memory. (iii)It was originally developed for nonlinear optimization problems with continuous variables. However, it is easily expanded to treat problems with discrete variables.

Pesodo code for basic Particle Swarm Optimization:
1. Initialize the ***swarm*** from the solution space
2. Evaluate ***fitness*** of individual ***particles***
3. Modify ***gbest***, ***pbest*** and ***velocity***
4. Move each ***particle*** to a new ***position***.
5. **Go to step 2**, and repeat until convergence or a stopping condition is satisfied.

Flow chart:

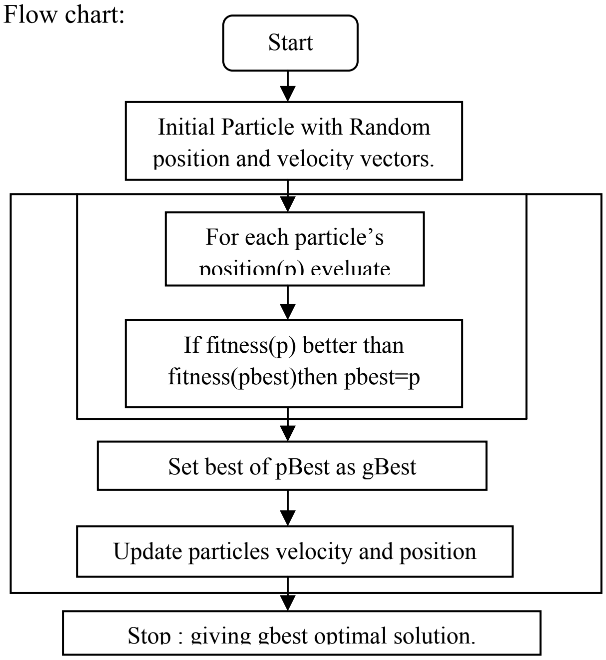


Fig 1 :Flow chart for Particle Swarm optimization

4.1 Position Updation of Particle

Based on velocity particle position is updated. From the Updation pbest and gbest are calculated according to QoS constraints.

$V_i^{k+1}=\omega V_i^k+ c_1 rand_1 \times (pbest_i- X_i^k) + c_2 rand_2 \times (gbest_i- X_i^k)$

$X_i^{k+1} = X_i^k + V_i^{k+1}$

Where,

V_i^k = Velocity of particle i at iteration k
V_i^{k+1}=Velocity of particle i at iteration k+1
ω=Inertia Weight between 0.9 to 0.1
c_j = Acceleration Coefficients j=1,2
$rand_i$ = random number between 0 and 1
X_i^k=Current position of particle i at iteration k
$pbest_i$=best position of particle i
gbest=position of best particle in a population
X_i^{k+1}=Position of the particle i at iteration k+1.

Finally best solution is obtained from the fitness values of each particle, maximum value is consider as a best optimal solution.

5. Proposed Work

Here, we proposed a work on some QoS Constraint. Like, Makespan, Cost and Deadline respectively. Finally we compare the result to the ACO with other works.

5.1 Makespan Constraints

The particle with the minimum time has the greater probability to be chosen. Initialization of particle with makespan as

$X_i^{k+1}=t_i \times p_j$
$Pbest_i = maximum(X_i^k)$
$Gbest = maximum(Pbest)$
Where t_i=Time of task,
p_j= Processor speed of resources.

5.2 Cost Constraints

The particle with the minimal cost has the greater probability to be chosen. Initialization of particle with cost as

$X_i^{k+1}=maximum(Cost_j)$
$Pbest_i = maximum(X_i^k)$
$Gbest = maximum(Pbest)$
Where $Cost_j$=Cost of each task,

5.3 Deadline Constraints

The particle with the minimal deadline has the greater probability to be chosen. Initialization of particle with deadline as

$$X_i^k = D_i - t_i$$

Where

t_i = time of each task,

D_i =Deadline of each task,

X_i^k = Number of satisfied task with deadline.

6. Load Balancing Problem

Load balancing has always been an issue since the emergence of distributed systems. In a distributed system there might be scenarios in which a task waits for a service at the queue of one resource, while at the same time another resource which is capable of serving the task is idle [10]. The purpose of a load balancing algorithm is to prevent these scenarios as much as possible. For parallel applications, load balancing attempts to distribute the computation load across multiple processors or machines as evenly as possible with the objective to improve performance. Generally, a load balancing scheme consists of three phases: information collection, decision making and data migration. During the information collection phase, the load balancer gathers the information of the distribution of workload and the state of computing environment and detects whether there is a load imbalance [11]. The decision making phase focuses on calculating an optimal data distribution, while the data migration phase transfers the excess amount of workload from one overloaded processor to another underloaded one.

See : (Fig : load balancing flow chart)

This techniques are implemented by the following formula,

$$R_l = Max\left\{ \sum_{i=1}^{n} t_{lj} \right\};\ l=1,2,3,\ldots.M$$

R_l = Position of the maximum Resource

M = No. of Resource

N = No. of task allocated to particular resource

$$R_k = Min\left\{ \sum_{j=1}^{n} t_{kj} \right\};\ k = 1,2,3,\ldots.M$$

R_k = Position of the minimum resource

M = No. of Resource

N = No. of task allocated to particular resource

T_m = Interchange min task.

T_p = Interchange max task.

$T_m = R_k$ { Min (tn)}

$T_p = R_l$ { Max (tn)}

7. Tables and Result

The proposed method is simulated for various combinations of heterogeneity of task and resources with different Matrix such as 128 x 8, 256 x 16, 512 x 32, and 1024 x 64. Here, 128, 256, 512, 1024 represents number of tasks and 8, 16, 32, 64 represents number of resources.The performance of both existing and proposed method is evaluated by the makespan, cost, deadline. The existing and proposed method is tested using some range of ETC matrix. The result was compared with different QoS such as Makespan, Cost and Deadline. The existing and proposed method is tested using some range of ETC matrix.

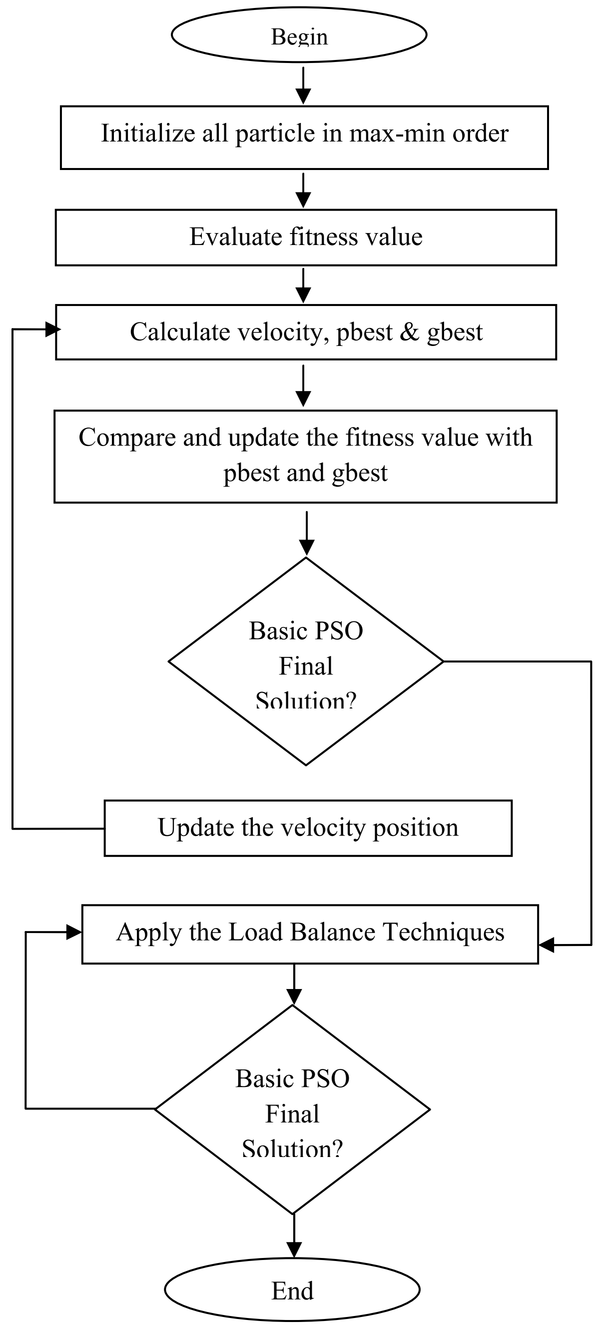
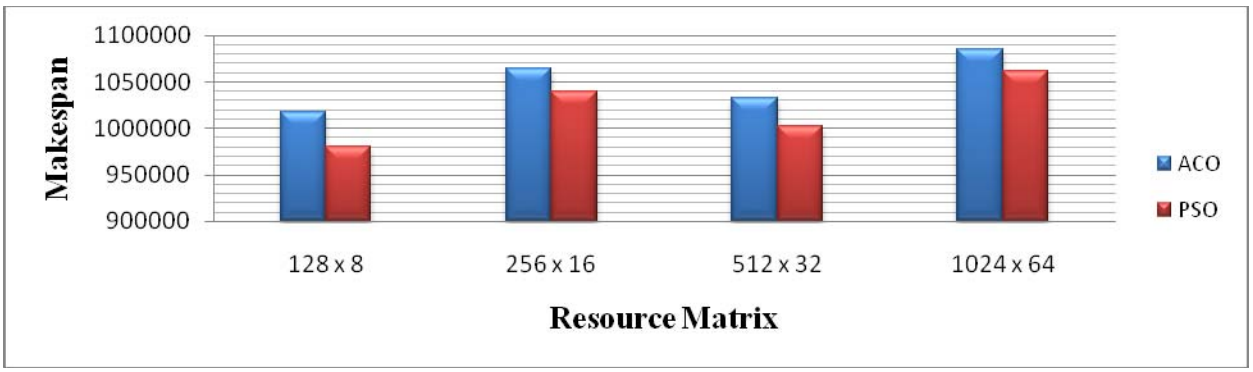


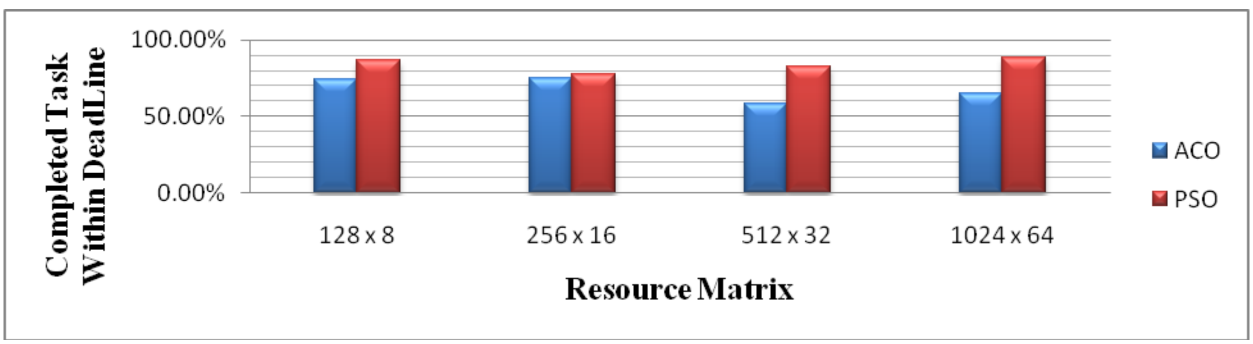
Fig 2: Flow chart Load Balancing Algorithm for PSO

IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 3, No 1, May 2012
ISSN (Online): 1694-0814
www.IJCSI.org

Table1 : High Task Length High Resource Speed High Hetrogenity Task Length High Hetrogenity Resource Speed

Scheduling Algorithm	Resource Matrix	Makespan(s)	Cost (Rs)	DeadLine
ACO	128 x 8	1017793.44	1932018.65	74.22%
PSO		980844.59	1480133.78	86.70%
ACO	256 x 16	1064909.53	1984450.25	75%
PSO		1039684.95	1927818.26	77.34%
ACO	512 x 32	1032971.56	1994438.15	58.39%
PSO		1002418.96	1895836.29	82.42%
ACO	1024 x 64	1084742.33	1947235.41	64.84%
PSO		1061161.71	1912811.93	88.18%





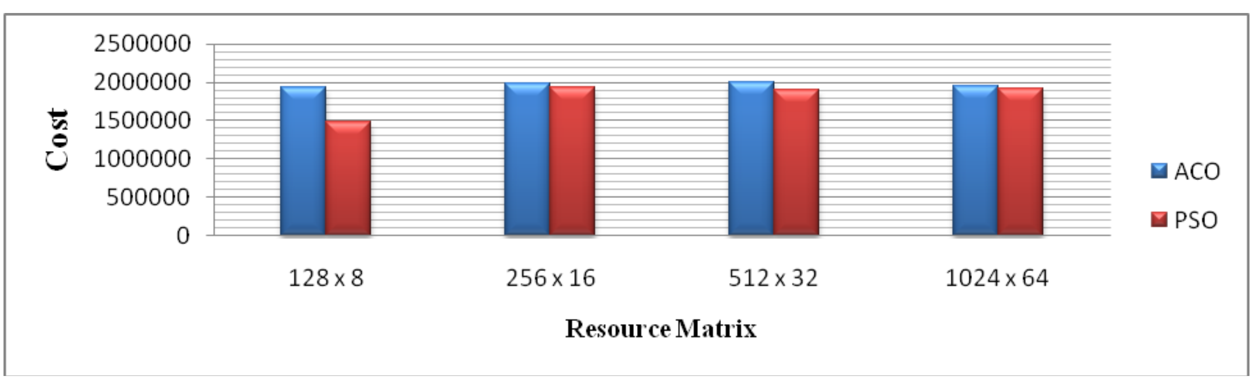


Table 2 : High Task Length High Resource Speed High Hetrogenity Task Length High Hetrogenity Resource Speed

Resource Matrix	PSO	MAX PSO	MIN PSO
128 x 8	980844.59	964924.6	978525.26
256 x 16	1039684.95	1001629.22	1019247.61
512 x 32	1002418.96	1007407.39	1026524.06
1024 x 64	1061161.71	1042027.49	1063068.99

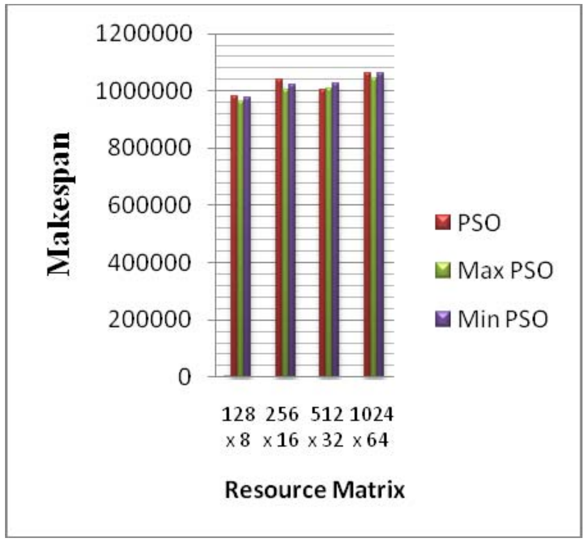


Table 3 : High Task Length High Resource Speed High Hetrogenity Task Length High Hetrogenity Resource Speed

Resource Matrix	PSO	MAX PSO	MAX PSO LB
128 x 8	980844.59	964924.6	894296.94
256 x 16	1039684.95	1001629.22	943012.86
512 x 32	1002418.96	1007407.39	921413.17
1024 x 64	1061161.71	1042027.49	925843.42

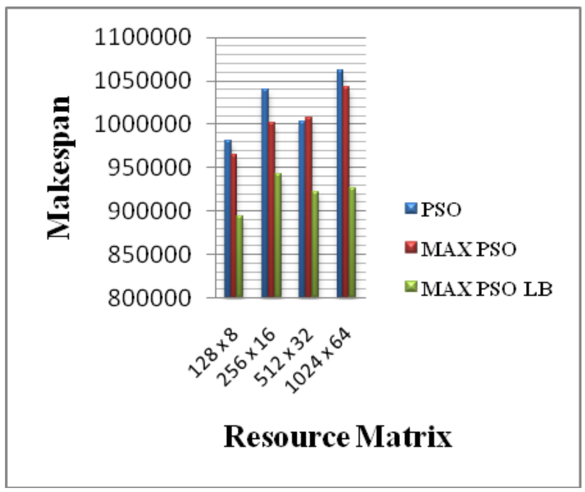
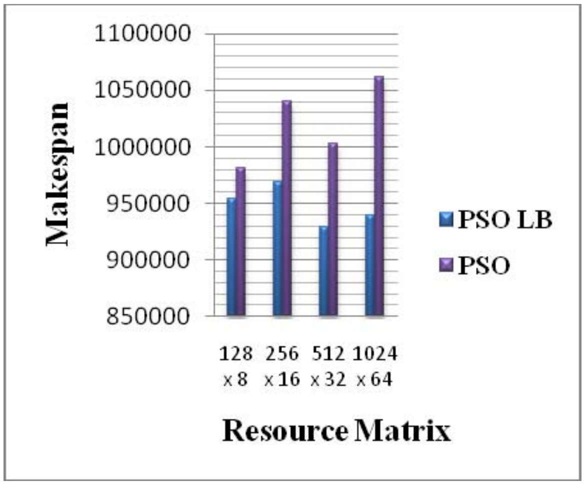


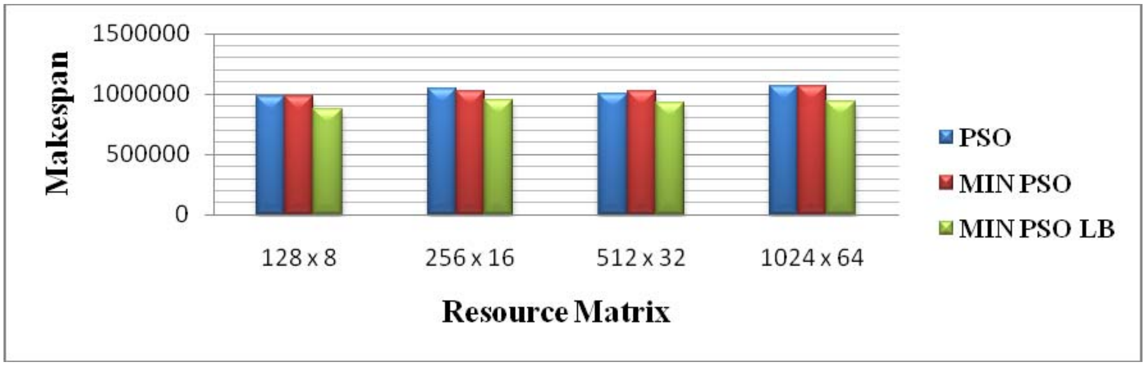
Table 4 : High Task Length High Resource Speed High Hetrogenity Task Length High Hetrogenity Resource Speed

Resource Matrix	PSO	PSO LB
128 x 8	980844.59	954027.75
256 x 16	1039684.95	968380.52
512 x 32	1002418.96	929024.05
1024 x 64	1061161.71	939735.21


1100000
1050000
1000000
950000
900000
850000
Makespan
128 x 8
256 x 16
512 x 32
1024 x 64
Resource Matrix
PSO LB
PSO

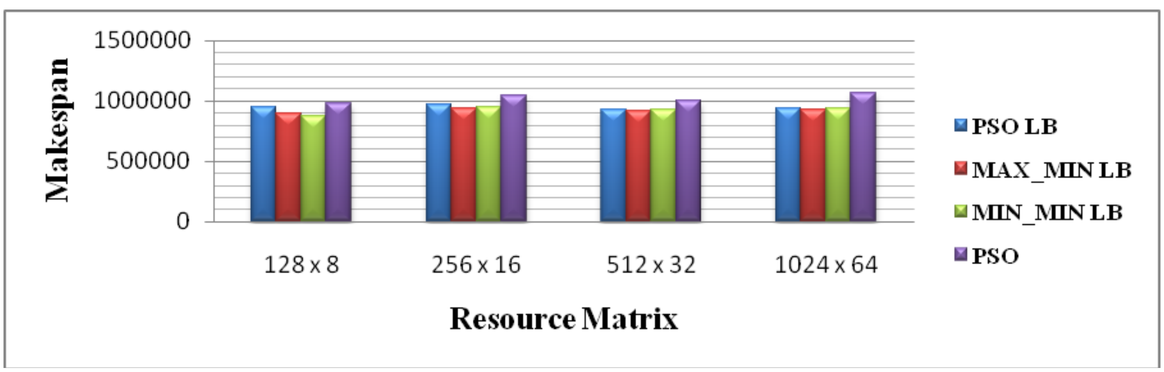
Resource Matrix	PSO	MIN PSO	MIN PSO LB
128 x 8	980844.59	978525.26	874566.22
256 x 16	1039684.95	1019247.61	945891.99
512 x 32	1002418.96	1026524.06	930246.91
1024 x 64	1061161.71	1063068.99	936800.34

Table 5 : High Task Length High Resource Speed High Hetrogenity Task Length High Hetrogenity Resource Speed



Resource Matrix	PSO	PSO LB	MAX PSO LB	MIN PSO LB
128 x 8	980844.59	954027.75	894296.94	874566.22
256 x 16	1039684.95	968380.52	943012.86	945891.99
512 x 32	1002418.96	929024.05	921413.17	930246.91
1024 x 64	1061161.71	939735.21	925843.42	936800.34

Table 6 : High Task Length High Resource Speed High Hetrogenity Task Length High Hetrogenity Resource Speed



8. Conclusion

Grid Scheduling is a critical design issue of grid computing. The major objective of grid scheduling is to reduce the makespan, cost and increase the number of tasks completed within deadline. The algorithm is developed based on PSO to find a proper resource allocation to jobs in Grid Environment. In this paper, Particle Swarm Optimization (PSO) Algorithm is developed based on three QoS makespan, cost and Deadline. Results show that PSO is better than ACO according to makespan, cost and Deadline. Load on resources are also balanced for the proposed PSO, Max PSO and Min PSO. Better makespan is achieved when the load on the resources are balanced, with Min-PSO gives the best result. Future work can focuses to compare these result to other swarm intelligence algorithm and then finding the better solution.

Reference

[1] I. Foster and C. Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1998.

[2] K. Q. Yan, S. C. Wang, C. P. Chang, and J. S. Lin. A hybrid load balancing policy un-derlying grid computing environment. Computer Standards and Interfaces, 29(2):161173, 2007.

[3] I. Foster and C. Kesselman. The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann Publishers, 1998.

[4] R. Schoonderwoerd,O.E. Holland, J. L. Bruten, and L. J. M. Rothkrantz. Ant-based load balancing in telecommunications networks. Adaptive Behavior, (2):169{207, 1996.

[5] C. Grosan, A. Abraham, and B. Helvik. Multiobjective evolutionary algorithms for scheduling jobs on computational grids. ADIS International Conference, Applied Com- puting, Salamanca, Spain, Nuno Guimares and Pedro Isaias 2007.

[6] M. Dorigo, V. Maniezzo, et A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Transactions on Systems, Man, and Cybernetics--Part B , volume 26, numéro 1, pages 29-41, 1996.

[7] E. Ridge, D. Kudenko, D. Kazakov, and E. Curry,"Moving Nature-Inspired Algorithms to Parallel, Asynchronous and Decentralised Environments," in Self-Organization and Autonomic Informatics (I), 2005, vol. 135, pp. 35-49.

[8] Shah-Hosseini, Hames (2009). "The intelligent water drops algorithm: a nature-inspired swarm-based optimization algorithm". International Journal of Bio-Inspired Computation 1 (1/2) : 71-79.

[9] Karaboga, Dervis (2010)Artificial bee colony algorithmScholarpedia, **5**(3): 6915.

[10] M. Livny and M. Melman. Load balancing in homogeneous broadcast distributed systems. SIGMETRICS Performance Evaluation Review, 11(1):47{55, 1981.

[11] Y. Li and Z. Lan. A Survey of Load Balancing in Grid Computing. Springer Berlin / Heidelberg, 2005.

[12] K. Kousalya and P. Balasubramanie," Task Severance and Task Parceling Based Ant Algorithm for Grid Scheduling", International journal of computational cognition (http://www.ijcc.us), vol. 7, no. 4, december 2009.

[13] D. Merkle, M. Middendorf, and H. Schmeck, "Ant colony optimization for resource-constrained project scheduling," IEEE Trans. Evol. Comput.,vol. 6, no. 4, Aug. 2002.

[14] Yarkhan .A, J. Dongarra, "Experiments with scheduling using simulated annealing in a grid environment", in: 3rd International Workshop on Grid Computing, 2002.

[15] Abraham, R. Buyya, and B. Nath." Nature's heuristics for scheduling jobs on computational Grids". In The 8th IEEE International Conference on Advanced Computing and Communications (ADCOM 2000), India, 2000.

[16] V. Di Martino, M. Mililotti, "Sub optimal scheduling in a grid using genetic algorithms", Parallel Computing 30 (2004) 553–565.

[17] Lei Zhang, Yuehui Chen, Bo Yang "Task Scheduling Based on PSO Algorithm in Computational Grid", 2006 Proceedings of the 6th International Conference on Intelligent Systems Design and Applications, vol-2, 16-18 Oct, 2006, Jinan, China.

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

[19] Shih-Tang Lo, Ruey-Maw Chen, Der-Fang Shiau and Chung-Lun Wu "Using Particle Swarm Optimization to Solve Resource-constrained Scheduling Problems" IEEE Conference on Soft Computing in Industrial Applications, 25-27 June, 2008, Muroran.

C.Kalpana MSc., MCA., M.Phil.,
She has completed her Post Graduate with M.Phil from Bharathiar University, Coimbatore. Now, she is working as a Assistant Professor in VLB Janaki Ammal Arts and Science College, Coimbatore. She presented 2 International Conference and 1 International workshop. Many national and international journals are also published. She is intersted in Grid computing, cloud computing, advanced networking. Now she is doing research papers in grid copmuting.

U.Karthick Kumar MSc., DWA., MCA., M.Phil., He is a Post Graduate with M.Phil from Bharathiar University, Coimbatore, Tamilnadu, India. He is working as a Assistant Professor in VLB Janaki Ammal Arts and Science College, Coimbatore. He has three years of experience in research. He presented paper in International, National Conference and Journals. His Interest areas are Grid Computing, Data Mining, Image Processing, Mobile Computing and Data Structures. Current research project Ant colony, PSO, Fair Scheduling, Sensor Network.